

**PENERAPAN SISTEM PAKAR UNTUK DIAGNOSIS AWAL PENYAKIT
HEPATITIS MENGGUNAKAN METODE *FORWARD CHAINING* DAN
*CERTAINTY FACTOR***



TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi Pada Program
Studi Sarjana Terapan Teknik Informatika

Oleh :

Nama : Mukhammad Sabda Abdulloh

NIM : 16090106

**PROGRAM STUDI SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA
TEGAL
2021**

HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Mukhammad Sabda Abdulloh

NIM : 16090106

Adalah mahasiswa Program Studi D IV Teknik Informatika Politeknik Harapan Bersama, dengan ini saya menyatakan bahwa laporan Tugas Akhir yang berjudul :

**“PENERAPAN SISTEM PAKAR UNTUK DIAGNOSIS AWAL
PENYAKIT HEPATITIS MENGGUNAKAN METODE *FORWARD*
CHAINING DAN *CERTAINTY FACTOR*”**

merupakan hasil pemikiran sendiri secara orisinil dan saya susun secara mandiri dengan tidak melanggar kode etik hak karya cipta. Pada laporan Tugas Akhir ini juga bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik tertentu di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari ternyata Laporan Tugas Akhir ini terbukti melanggar kode etik karya cipta atau merupakan karya yang dikategorikan mengandung unsur plagiarisme, maka saya bersedia untuk melakukan penelitian baru dan menyusun laporannya sebagai laporan Tugas Akhir, sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya dan sesungguhnya.

Tegal, Oktober 2021

Yang membuat pernyataan



Mukhammad Sabda Abdulloh

NIM. 16090106

HALAMAN REKOMENDASI

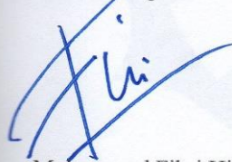
Pembimbing Tugas Akhir memberikan rekomendasi kepada:

Nama : Mukhammad Sabda Abdulloh
NIM : 16090106
Program Studi : D IV Teknik Informatika
Judul Tugas Akhir : Penerapan Sistem Pakar Untuk Diagnosis Awal Penyakit
Hepatitis Menggunakan Metode *Forward Chaining* Dan
Certainty Factor

Mahasiswa tersebut telah layak dinyatakan lulus menyelesaikan bimbingan dan dapat mengikuti Ujian Tugas Akhir pada tahun akademik 2020/2021

Tegal, Juli 2021

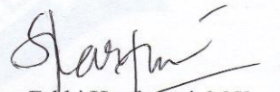
Pembimbing I



Muhammad Fikri Hidayattullah, S.T., M.Kom

NIPY. 09.016.307

Pembimbing II



Sharfina Febbi Handayani, M.Kom

NIPY. 08.020.451

HALAMAN PENGESAHAN


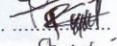
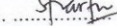
Nama : Mukhammad Sabda Abdulloh
NIM : 16090106
Program Studi : D IV Teknik Informatika
Judul Tugas Akhir : Penerapan Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis Menggunakan Metode *Forward Chaining* Dan *Certainty Factor*
Dinyatakan LULUS/TIDAK LULUS setelah dipertahankan di hadapan Dewan Penguji Tugas Akhir Program Studi D IV Teknik Informatika Politeknik Harapan Bersama

Tegal, November 2021

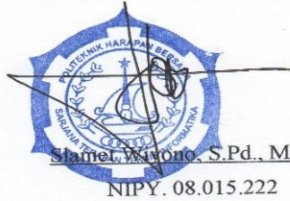
Dewan Penguji :

- Nama
1. Ketua : Dyah Apriliani, S.T., M.Kom
 2. Anggota I : Rizki Wijayatun Pratiwi, M.Cs
 3. Anggota II : Sharfina Febbi Handayani, M.Kom

Tanda Tangan

1. 
2. 
3. 

Mengetahui,
Ketua Program Studi D IV Teknik Informatika


Shamel, Kunto, S.Pd., M.Eng
NIPY. 08.015.222

ABSTRAK

Indonesia merupakan salah satu negara dengan penderita penyakit dalam terbanyak, dan salah satu penyakit dalam yang banyak di derita oleh masyarakat Indonesia adalah penyakit Hepatitis. Hepatitis adalah kelainan pada hati berupa peradangan pada hati. Peradangan ditandai dengan peningkatan kadar enzim hati, yang disebabkan oleh berbagai faktor seperti faktor infeksi dan non infeksi. Dan untuk mengetahui gejala awal atau gejala fisik seseorang terkena penyakit hepatitis harus segera melakukan pemeriksaan di instalasi kesehatan terdekat, untuk selanjutnya melakukan tes Laboratorium yang memerlukan waktu khusus dan biaya. Sistem pakar ini dikembangkan dengan menggunakan Framework Laravel serta menggunakan metode pengembangan *System Development Life Cycle (SDLC)* dan menggunakan metode penalaran *Forward Chaining* dan *Certainty Factor*. Diharapkan dengan pembuatan sistem pakar untuk diagnosis awal penyakit hepatitis A, B, dan C yang berbasis *web* yang diharapkan memudahkan pengguna yang masih awam terhadap pengetahuan maupun informasi terkait gejala-gejala hepatitis yang mudah menular untuk dapat mengetahui gejalanya diawal, meskipun pada akhirnya penanganan medis sebagai proses utama penyembuhan dengan keahlian dari dokter dibidangnya.

Kata kunci : sistem pakar, *forward chaining*, *certainty factor*, penyakit hepatitis

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, Tuhan Yang Maha Pengasih dan Penyayang yang telah melimpahkan segala rahmat, hidayah, dan inayah-Nya hingga terselesaikannya laporan Tugas Akhir yang berjudul **“PENERAPAN SISTEM PAKAR UNTUK DIAGNOSIS AWAL PENYAKIT HEPATITIS MENGGUNAKAN METODE *FORWARD CHAINING* DAN *CERTAINTY FACTOR*”**.

Tugas Akhir merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat kelulusan dalam mencapai derajat Sarjana Sains Terapan pada Program Studi D IV Teknik Informatika Politeknik Harapan Bersama. Selama melaksanakan penelitian ini dan kemudian tersusun dalam laporan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan, dukungan, dan bimbingan.

Pada kesempatan ini, tak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Nizar Suhendra, S.E., MPP selaku Direktur Politeknik Harapan Bersama Tegal.
2. Slamet Wiyono, S.Pd., M.Eng selaku Ketua Program Studi D IV Teknik Informatika Politeknik Harapan Bersama Tegal.
3. Muhammad Fikri Hidayattullah, S.T., M.Kom selaku Pembimbing I.
4. Sharfina Febbi Handayani, M.Kom selaku Pembimbing II.
5. Ahli Penyakit Dalam dr.Yit.Sp, PD selaku Narasumber.
6. Semua pihak yang telah mendukung, membantu serta mendoakan penyelesaian laporan Tugas Akhir ini.

Semoga laporan Tugas Akhir ini dapat memberikan sumbangan untuk pengembangan ilmu pengetahuan dan teknologi.

Tegal, Oktober 2021

Penulis,

Mukhammad Sabda Abdulloh

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN	ii
HALAMAN REKOMENDASI	iii
HALAMAN PENGESAHAN.....	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xvii
BAB I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Pembatasan Masalah	3
1.4. Tujuan.....	4
1.5. Manfaat Penelitian.....	4
1.6. Metodologi Penelitian	5
1.7. Sistematika Penulisan.....	7
BAB II. TINJAUAN PUSTAKA.....	10
BAB III. LANDASAN TEORI.....	16

3.1. Sistem Pakar	16
3.1.1 Pengertian Sistem Pakar	16
3.1.2 Kelebihan dan Kekurangan Sistem Pakar	16
3.2. Metode Forward Chaining	17
3.3. Metode Certainty Factor (Faktor Kepastian).....	19
3.4. Definisi Hepatitis.....	22
3.4.1 Hepatitis A.....	23
3.4.2 Hepatitis B.....	25
3.4.3 Hepatitis C.....	26
3.5. Mysql.....	28
3.6. PHP.....	28
3.7. Codeigniter	29
3.8. Unified Modeling Language (UML).....	29
3.9. Black Box Testing.....	37
BAB IV. DESAIN DAN PERANCANGAN.....	38
4.1. Perancangan Sistem.....	38
4.2. Perancangan UML(Unified Modeling Language).....	38
4.2.1. Usecase Diagram.....	38
4.2.2. Activity Diagram.....	41
4.2.3. Sequence Diagram	51
4.2.4. Class Diagram	61
4.3 Perancangan Database.....	62
4.4 Perancangan Antar Muka	64

4.5 Perancangan Sistem Pakar	76
4.5.1 Tabel Keputusan.....	76
4.5.2 Pohon Keputusan.....	77
4.5.3 Kaidah Produksi	78
BAB V. HASIL DAN PEMBAHASAN.....	79
5.1. Hasil Penelitian.....	79
5.1.1. Implementasi Antar Muka	79
5.1.2. Black Box Testing	97
5.1.3. Validasi Sistem	106
5.2. Pembahasan	109
5.2.1. Penerapan Perhitungan Algoritma Certainty Factor	109
BAB VI. PENUTUP	129
6.1. Kesimpulan.....	129
6.2. Saran.....	129
DAFTAR PUSTAKA	130

DAFTAR TABEL

	Halaman
Tabel 3.1 Tabel Penyakit.....	14
Tabel 3.2 Tabel Keputusan	14
Tabel 3.3 Tabel Pembentukan Rule	15
Tabel 3.4 Tabel Input Pakar	17
Tabel 3.5 Tabel Input User	17
Tabel 3.6 Tabel Nilai CF Kombinasi	17
Tabel 3.8 Tabel Use Case Diagram.....	27
Tabel 3.9 Tabel Sequence Diagram	29
Tabel 3.10 Tabel Activity Diagram	30
Tabel 3.11 Tabel Class Diagram.....	31
Tabel 4.1 Identifikasi Aktor	34
Tabel 4.2 Identifikasi Use Case Diagram User.....	34
Tabel 4.3 Identifikasi User Case Diagram Admin	35
Tabel 4.4 Struktur Tabel Admin	57
Tabel 4.5 Struktur Tabel Berita.....	57
Tabel 4.6 Struktur Tabel Diagnosis	58
Tabel 4.7 Struktur Tabel Kategori	58
Tabel 4.8 Struktur Tabel Kuisisioner.....	58
Tabel 4.9 Struktur Tabel Rekam Diagnosis	58
Tabel 4.10 Struktur Tabel User.....	58
Tabel 4.11 Tabel Keputusan	71

Tabel 5.1 Pengujian Halaman Login Admin.....	93
Tabel 5.2 Pengujian Halaman Data Admin.....	94
Tabel 5.3 Pengujian Halaman Kuisisioner (Admin).....	95
Tabel 5.4 Pengujian Halaman Diagnosis (Admin)	96
Tabel 5.5 Pengujian Halaman Berita (Admin).....	97
Tabel 5.6 Pengujian Halaman Kategori Berita (Admin).....	98
Tabel 5.7 Pengujian Halaman Login User	99
Tabel 5.8 Pengujian Halaman Daftar Akun (User).....	100
Tabel 5.9 Validasi Sistem	101
Tabel 5.10 CF Input Pakar Hepatitis A.....	104
Tabel 5.11 CF Input User Hepatitis A	105
Tabel 5.12 CF Kombinasi Hepatitis A.....	105
Tabel 5.13 CF Input Pakar Hepatitis B	107
Tabel 5.14 CF Input User Hepatitis B.....	108
Tabel 5.15 CF Kombinasi Hepatitis B	108
Tabel 5.16 CF Input Pakar Hepatitis C	110
Tabel 5.17 CF Input User Hepatitis C.....	111
Tabel 5.18 CF Kombinasi Hepatitis C	111

DAFTAR GAMBAR

	Halaman
Gambar 4.1 Use Case Diagram Sistem Pakar Diagnosis Hepatitis	33
Gambar 4.2 Activity Diagram Login (User)	36
Gambar 4.3 Activity Diagram Beranda (User)	37
Gambar 4.4 Activity Diagram Diagnosis (User).....	37
Gambar 4.5 Activity Diagram Berita (User).....	38
Gambar 4.6 Activity Diagram Profil (User)	38
Gambar 4.7 Activity Diagram Kontak (User).....	39
Gambar 4.8 Activity Diagram Akun Saya (User).....	39
Gambar 4.9 Activity Diagram Hasil Diagnosis (User)	40
Gambar 4.10 Activity Diagram Logout (User)	40
Gambar 4.11 Activity Diagram Beranda (Admin).....	41
Gambar 4.12 Activity Diagram Login (Admin)	41
Gambar 4.13 Activity Diagram User (Admin)	42
Gambar 4.14 Activity Diagram Admin (Admin).....	42
Gambar 4.15 Activity Diagram Kuisisioner (Admin)	43
Gambar 4.16 Activity Diagram Diagnosis (Admin).....	43
Gambar 4.17 Activity Diagram Berita (Admin)	44
Gambar 4.18 Activity Diagram Kategori Berita (Admin)	44
Gambar 4.19 Activity Diagram Rekam Diagnosis (Admin).....	45
Gambar 4.20 Activity Diagram Logout (Admin)	45
Gambar 4.21 Sequence Diagram Login (User).....	46

Gambar 4.22 Sequence Diagram Beranda (User)	47
Gambar 4.23 Sequence Diagram Berita (User).....	47
Gambar 4.24 Sequence Diagram Diagnosis (User)	48
Gambar 4.25 Sequence Diagram Profil (User)	48
Gambar 4.26 Sequence Diagram Kontak (User).....	49
Gambar 4.27 Sequence Diagram Akun Saya (User).....	49
Gambar 4.28 Sequence Diagram Hasil Diagnosis (User).....	50
Gambar 4.29 Sequence Diagram Logout (User).....	50
Gambar 4.30 Sequence Diagram Login (Admin)	51
Gambar 4.31 Sequence Diagram Beranda (Admin)	51
Gambar 4.32 Sequence Diagram User (Admin)	52
Gambar 4.33 Sequence Diagram Admin (Admin).....	52
Gambar 4.34 Sequence Diagram Kuisiner (Admin).....	53
Gambar 4.35 Sequence Diagram Diagnosis (Admin).....	53
Gambar 4.36 Sequence Diagram Berita (Admin).....	54
Gambar 4.37 Sequence Diagram Kategori Berita (Admin).....	54
Gambar 4.38 Sequence Diagram Rekam Diagnosis (Admin)	55
Gambar 4.39 Sequence Diagram Logout (Admin)	55
Gambar 4.40 Class Diagram	56
Gambar 4.41 Rancangan Halaman Beranda (User)	59
Gambar 4.42 Rancangan Halaman Login (User).....	60
Gambar 4.43 Rancangan Halaman Daftar (User)	60
Gambar 4.44 Rancangan Halaman Diagnosis (User)	61

Gambar 4.45 Rancangan Halaman Berita (User).....	61
Gambar 4.46 Rancangan Halaman Profil (User)	62
Gambar 4.47 Rancangan Halaman Hasil Diagnosis (User).....	62
Gambar 4.48 Rancangan Halaman Akun Saya (User).....	63
Gambar 4.49 Rancangan Halaman Kontak (User).....	63
Gambar 4.50 Rancangan Halaman Login (Admin)	64
Gambar 4.51 Rancangan Halaman Beranda (Admin).....	64
Gambar 4.52 Rancangan Halaman Admin (Admin).....	65
Gambar 4.53 Rancangan Halaman Tambah Admin (Admin).....	65
Gambar 4.54 Rancangan Halaman User (Admin)	66
Gambar 4.55 Rancangan Halaman Kuisisioner (Admin)	66
Gambar 4.56 Rancangan Halaman Tambah Kuisisioner (Admin).....	67
Gambar 4.57 Rancangan Halaman Diagnosis (Admin).....	67
Gambar 4.58 Rancangan Halaman Tambah Diagnosis (Admin).....	68
Gambar 4.59 Rancangan Halaman Rekam Diagnosis (Admin).....	68
Gambar 4.60 Rancangan Halaman Berita (Admin)	69
Gambar 4.61 Rancangan Halaman Tambah Berita (Admin).....	69
Gambar 4.62 Rancangan Halaman Kategori Berita (Admin)	70
Gambar 4.63 Rancangan Halaman Tambah Kategori Berita (Admin)	70
Gambar 4.64 Pohon Keputusan Sistem Pakar Diagnosis Hepatitis	72
Gambar 5.1 Tampilan Halaman Beranda (User).....	75
Gambar 5.2 Controller Halaman Beranda User	75
Gambar 5.3 Tampilan Halaman Login (User)	76

Gambar 5.4 Controller Halaman Login (User)	76
Gambar 5.5 Tampilan Halaman Diagnosis (User).....	77
Gambar 5.6 Controller Halaman Diagnosis (User).....	77
Gambar 5.7 Tampilan Halaman Berita (User)	78
Gambar 5.8 Controller Halaman Berita (User)	78
Gambar 5.9 Tampilan Halaman Kontak (User)	79
Gambar 5.10 View Halaman Kontak (User).....	79
Gambar 5.11 Tampilan Halaman Profil (User).....	80
Gambar 5.12 View Tampilan Halaman Profil (User)	80
Gambar 5.13 Tampilan Halaman Akun Saya (User)	81
Gambar 5.14 Controller Halaman Akun Saya (User)	81
Gambar 5.15 Tampilan Halaman Hasil Diagnosis (User)	82
Gambar 5.16 Controller Halaman Hasil Diagnosis (User)	82
Gambar 5.17 Tampilan Halaman Login (Admin).....	83
Gambar 5.18 Controller Halaman Login (Admin).....	83
Gambar 5.19 Tampilan Halaman Beranda (Admin)	84
Gambar 5.20 Controller Halaman Beranda (Admin).....	84
Gambar 5.21 Tampilan Halaman User (Admin).....	85
Gambar 5.22 Controller Halaman User (Admin).....	85
Gambar 5.23 Tampilan Halaman Admin (Admin)	86
Gambar 5.24 Controller Halaman Admin (Admin)	86
Gambar 5.25 Tampilan Halaman Kuisisioner (Admin).....	87
Gambar 5.26 Controller Halaman Kuisisioner (Admin)	87

Gambar 5.27 Tampilan Halaman Diagnosis (Admin)	88
Gambar 5.28 Controller Halaman Diagnosis (Admin)	88
Gambar 5.29 Tampilan Halaman Rekam Diagnosis (Admin).....	89
Gambar 5.30 Controller Halaman Rekam Diagnosis (Admin).....	89
Gambar 5.31 Tampilan Halaman Berita (Admin).....	90
Gambar 5.32 Controller Halaman Berita (Admin)	90
Gambar 5.33 Tampilan Halaman Kategori Berita (Admin).....	91
Gambar 5.34 Controller Halaman Kategori Berita (Admin)	92
Gambar 5.35 Penerapan Rules 1	102
Gambar 5.37 Penerapan Rules 2	103
Gambar 5.38 Penerapan Rules 3	103

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Kesepakatan Bimbingan Tugas Akhir Pembimbing 1	A-1
Lampiran 2 Surat Kesepakatan Bimbingan Tugas Akhir Pembimbing 2	A-2
Lampiran 3 Lembar Bimbingan Tugas Akhir Pembimbing 1	B-1
Lampiran 4 Lembar Bimbingan Tugas Akhir Pembimbing 2	B-2
Lampiran 5 Surat Permohonan Ijin Penelitian	C-1

BAB I

PENDAHULUAN

1.1 Latar Belakang.

Penyakit Hepatitis merupakan masalah kesehatan masyarakat dinegara berkembang didunia, termasuk Indonesia. Menurut hasil Riset Kesehatan Dasar (Riskesdas) tahun 2018 yang dikeluarkan oleh Kementerian kesehatan, jenis hepatitis yang banyak menginfeksi penduduk Indonesia adalah Hepatitis B (21,8%), Hepatitis A (19,3%), dan Hepatitis C (2,5%) [1].

Sedangkan menurut hasil riset WHO (*World Health Organization*) tahun 2020 menyebutkan bahwa hepatitis menjadi salah satu penyakit dengan jumlah terbanyak ke-2 didunia sebagai penyakit yang disebabkan oleh virus. Di mana ditengah pandemi, penyebaran virus hepatitis juga tak berhenti. Selain itu, banyak kasus hepatitis yang tidak terdiagnosis. Hal ini menyebabkan penanganan yang semakin sulit dikarenakan banyak orang yang tidak sadar bahwa dirinya menderita hepatitis dan tidak segera mendapat perawatan [2].

Hepatitis sendiri adalah peradangan hati yang bisa berkembang menjadi fibrosis (jaringan parut), sirosis atau kanker hati. Hepatitis disebabkan oleh berbagai faktor seperti infeksi virus, zat beracun (misalnya alkohol, obat-obatan tertentu), dan penyakit autoimun [3]. Beberapa tipe hepatitis dapat sembuh dengan sendirinya tanpa mengakibatkan kondisi yang lebih serius. Meski demikian, ada tipe hepatitis lain yang membutuhkan pengobatan segera. Sebab tanpa pengobatan segera, hepatitis tersebut bisa menyebabkan jaringan parut pada hati (sirosis hati), yang akan berakhir dengan gagal hati atau dalam beberapa kasus, menjadi kanker hati [4].

Salah satu aspek yang sering memanfaatkan bantuan teknologi adalah aspek kesehatan. Para ahli terus melakukan riset dan pengembangan teknologi kedokteran, salah satunya di bidang *Artificial Intelligence (AI)* yang mengarah pada sistem pakar. Sistem pakar sendiri merupakan sistem

yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli, atau dengan kata lain sistem pakar adalah sistem yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli [5]. Banyak metode yang dapat digunakan dalam pengembangan sistem pakar, seperti *Analytical Hierarchy Process*, *Depth-First Search*, *Breadth-First Search*, *Best-First Search*, *Forward Chaining*, *Backward Chaining*, *Certainty Factor*, *Dempster-Shafer* dan lain-lain. Untuk menentukan suatu ketidakpastian dalam hal ini adalah mendiagnosis penyakit hepatitis berdasarkan gejala yang di-input-kan *user*, metode yang baik adalah *Certainty Factor* dan *Forward Chaining*. Pada penelitian yang dilakukan oleh Doddy Teguh Yuwono [6], menggabungkan metode *Certainty Factor* dan *Forward Chaining* untuk mendiagnosis hama anggrek *Coelogyne Pandurata*. Hasil penerapan Metode *Forward Chaining* dan *Certainty Factor* dapat memberikan diagnosis hama pada Anggrek *Coelogyne Pandurata* berdasarkan gejala-gejala yang diberikan. Berdasarkan hasil perhitungan, maka keterangan tingkat keyakinan berdasarkan tabel interpretasi dari pakar dan persentase akhir sebesar 93,0736% adalah sangat mungkin kedua metode ini diterapkan untuk menyelesaikan masalah yang ada. Untuk itu, pada penelitian yang akan dilakukan yaitu membangun sistem pakar dengan menggunakan metode *Certainty Factor* dan *Forward Chaining*.

Metode *Certainty Factor* memiliki keakuratan yang lebih karena cara perhitungannya hanya dapat membandingkan tiap dua nilai saja [6]. Sedangkan, *Forward Chaining* merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (*conclusion*) dari fakta tersebut. *Forward chaining* bisa dikatakan sebagai strategi inferensi yang bermula dari sejumlah fakta yang diketahui. Pencarian dilakukan dengan menggunakan aturan yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga goal

dicapai atau hingga sudah tidak ada *rules* lagi yang premis nya cocok dengan fakta yang diketahui maupun fakta yang diperoleh [7].

Permasalahan yang muncul adalah terbatasnya jumlah, waktu dan tenaga dari seorang dokter sehingga untuk melakukan konsultasi ketika dokter berhalangan hadir akan menyulitkan pasien. Untuk itu, seorang dokter membutuhkan asisten dokter dalam menangani penyakit tersebut, sehingga kebutuhan pasien untuk mendapatkan pelayanan medis yang lebih baik dapat terpenuhi. Selain itu, bagi masyarakat pada umumnya yang membutuhkan informasi tentang penyakit hepatitis mulai dari gejala yang terjadi, penentuan jenis penyakit sampai dengan solusi untuk mengatasi penyakit tersebut masih bergantung kepada dokter. Selain jumlah dan tenaga dokter yang terbatas, biaya yang dibutuhkan untuk berkonsultasi dengan dokter tidak sedikit.

Untuk itu, berdasarkan kebutuhan seorang asisten dokter dan masyarakat umum maka perlu dibuat aplikasi sistem pakar berbasis web untuk melakukan diagnosis awal penyakit hepatitis. Di mana dalam aplikasi ini memberikan suatu pernyataan dan informasi kepada sistem pakar, sehingga dapat mengambil kesimpulan dengan cepat dan tepat. Aplikasi Sistem Pakar Diagnosis Awal Penyakit Hepatitis diharapkan mampu memberikan solusi sedini mungkin dan proses penanganan terhadap pasien bisa dilakukan dengan cepat.

1.2 Rumusan Masalah.

Berdasarkan latar belakang masalah, maka permasalahan yang dapat dirumuskan sebagai berikut :

1. Bagaimana merancang dan membuat sistem pakar dengan metode *Certainty Factor* yang dapat mendiagnosis gejala hepatitis ?
2. Seberapa tinggi akurasi metode *Certainty Factor* dalam mendiagnosis penyakit hepatitis ?

1.3 Pembatasan Masalah

Adapun batasan dari penelitian yang dilakukan adalah sebagai berikut :

1. Pengembangan sistem pakar hanya terbatas untuk mendiagnosis atau mendeteksi penyakit Hepatitis A, Hepatitis B dan Hepatitis C berdasarkan gejala yang ditunjukkan oleh program
2. Pengisian gejala dan nilai CF gejala terhadap penyakit diperoleh dari pakar atau dokter dan buku-buku.
3. Metode yang digunakan untuk mengatasi ketidakpastian nya menggunakan metode *certainty factor*.
4. *Output* berupa jenis penyakit serta solusi atas penyakit tersebut.
5. Aplikasi berbasis *web* serta dikembangkan menggunakan bahasa pemrograman PHP dengan *Codeigniter* sebagai *Framework* nya.

1.4 Tujuan

Tujuan dari penelitian ini adalah merancang dan mengaplikasikan sistem pakar yang mampu menentukan diagnosis awal penyakit hepatitis berdasarkan pada gejala-gejala yang dialami oleh pasien dengan memperhatikan aturan-aturan secara cepat dan tepat.

1.5 Manfaat Penelitian

1. Bagi Paramedis
 - a. Penelitian ini bermanfaat untuk membantu paramedis dalam hal ini antara lain asisten dokter penyakit dalam mendiagnosis penyakit Hepatitis A, Hepatitis B, dan Hepatitis C.
 - b. Memudahkan dokter penyakit dalam dan pasien untuk menentukan jenis penyakit hepatitis berdasarkan gejala-gejala yang ada dan didapat solusi yang tepat.
2. Bagi Masyarakat
 - a. Membantu masyarakat dalam mengetahui informasi tentang penyakit hepatitis, sehingga dapat dicegah agar tidak terjadi penyakit.
 - b. Memudahkan masyarakat dalam mendiagnosis penyakit hepatitis secara cepat dan tepat tanpa menunggu kehadiran dokter.

3. Bagi Penulis
 - a. Penulis dapat menerapkan teori-teori yang telah diperoleh selama berada dibangku perkuliahan ke dalam kegiatan langsung dunia kerja.
 - b. Meningkatkan kemampuan mahasiswa untuk menganalisis suatu masalah ke dalam sebuah sistem sehingga mampu membuat aplikasi yang sesuai.
4. Bagi Politeknik Harapan Bersama
 - a. Sebagai salah satu wujud pengabdian dan kepedulian sekolah tinggi kepada masyarakat
 - b. Tugas Akhir merupakan kegiatan umpan balik yang berfungsi untuk mengetahui apakah sekolah tinggi mampu memberikan pengetahuan kepada para mahasiswanya agar mampu memenuhi segala sesuatu dengan tuntutan dunia kerja.

1.6 Metodologi Penelitian

1. Identifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini adalah bagaimana membuat sistem pakar untuk diagnosis awal penyakit hepatitis berdasarkan gejala yang dirasakan oleh pasien.

2. Jenis Data

Jenis data dapat berupa catatan-catatan dalam kertas, buku atau tersimpan dalam *file* di *database*, serta wawancara. Dalam hal ini penulis menggunakan jenis data berupa sumber data primer dan sumber data sekunder.

a. Data Primer

Data primer adalah data yang didapat dari sumber pertama, baik dari individu atau perseorangan yang diperoleh secara langsung di lapangan oleh peneliti sebagai objek penulisan. Wawancara dengan penggunaan pedoman (*interview guide*) dimaksudkan untuk wawancara yang lebih mendalam dengan memfokuskan pada persoalan-persoalan yang akan diteliti. Dalam

penelitian yang ini termasuk dalam data primer adalah data yang diperoleh dari dokter melalui wawancara secara langsung, yang meliputi data jenis penyakit hepatitis, gejala penyakit hepatitis dan solusinya.

b. Data Sekunder

Dalam menyelesaikan laporan ini, data diperoleh dari media pustaka tentang teori-teori sistem yang berkaitan dengan makalah yang diteliti, merupakan informasi tambahan yang mendukung data dari seorang dokter yang meliputi data penyakit hepatitis.

3. Metode Pengumpulan Data

Salah satu masalah yang terpenting dalam penelitian adalah melalui metode tertentu untuk memecahkan suatu masalah yang diperoleh dengan tujuan agar mendapat hasil yang dapat dipertanggungjawabkan. Adapun langkah-langkah dalam teknik pengumpulan data suatu penelitian adalah sebagai berikut:

a. Observasi Lingkungan

Tahap observasi lingkungan merupakan tahap paling awal didalam kegiatan penelitian ini. Pada tahap ini akan dilakukan identifikasi masalah dengan penyakit Hepatitis A, Hepatitis B, dan Hepatitis C. Pengumpulan data yang dilakukan melalui pengamatan, pencatatan dan pencarian tentang gejala penyakit hepatitis dan solusinya.

b. Studi Pustaka

Dalam mempelajari data manual dan referensi melalui buku-buku acuan yang berhubungan dengan masalah yang dihadapi. Dalam tahap ini akan diperoleh data penyakit hepatitis data dari seorang dokter/pakar.

c. Wawancara

Teknik wawancara adalah suatu metode pengumpulan data dengan bertanya langsung kepada responden, sehingga diperoleh data yang akurat. Dalam metode ini penulis mengadakan tanya

jawab kepada seorang pakar yang berkenaan dengan kebutuhan data yang akan diperoleh. Metode ini bertujuan untuk memperoleh data yang akurat karena pakar/dokter dapat memberikan masukan berdasarkan pada penanganan penyakit yang sedang diderita. Wawancara dilakukan dengan Ahli Penyakit Dalam dr.Yit.Sp, PD di Klinik Utama Kualita Medica Jl. Raya Utara Adiwerna No. 113 Kabupaten Tegal pada tanggal 21 April 2021. Hasil dari wawancara tersebut diperoleh data berupa data gejala hepatitis, tingkat kepastian dari setiap gejala yang muncul, pencegahan hepatitis, serta pengobatan hepatitis. Dan selanjutnya data tersebut dianalisis untuk mengolah data menjadi informasi sehingga dapat difahami dan berkaitan dengan penelitian.

1.7 Sistematika Penulisan

Sistematika penulisan laporan ini dibagi dalam beberapa bab dengan pokok permasalah nya. Sistematika penulisan secara umum dari laporan ini sebagai berikut:

1. Bagian Awal Laporan

Bagian awal penulisan laporan berupa Halaman Sampul Depan, Halaman Judul, Halaman Pernyataan, Halaman Rekomendasi, Halaman Pengesahan Abstrak, Kata Pengantar, Daftar Isi, Daftar Tabel, Daftar Gambar, Dan Daftar Lampiran.

2. Bagian Utama Laporan Tugas Akhir

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang diambilnya judul “Penerapan Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis Menggunakan Metode *Forward Chaining* Dan *Certainty Factor*”, perumusan masalah, pembatasan masalah, tujuan, manfaat, metodologi penelitian, dan sistematika penulisan yang menjelaskan secara garis besar substansi yang diberikan masing-masing bab.

BAB II : TINJAUAN PUSTAKA

Bab ini menguraikan dari latar belakang yang berkaitan dengan penelitian-penelitian serupa yang pernah dilakukan yang dibahas secara kritis. Berisi penjelasan tentang inti sari latar belakang, tujuan, metode, dan hasil penelitian

BAB III : LANDASAN TEORI

Dalam bab ini tentang landasan teori yang akan digunakan untuk penyelesaian laporan penelitian yang berkaitan dengan pembuatan Aplikasi Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis Menggunakan Metode *Forward Chaining* Dan *Certainty Factor*.

BAB IV : PERANCANGAN DAN DESAIN

Bab ini menguraikan secara detail rancangan dan desain terhadap penelitian yang dilakukan dari sistem yang dibangun mengani data-data yang dibuat dalam suatu perancangan User Interface dan perancangan *Unidentified Modelling Language (UML)*, rancangan ini terdiri dari: *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

BAB V : HASIL DAN PEMBAHASAN PENELITIAN

Dalam bab ini berisi tentang tangkapan layar hasil penelitian dan hasil penelitian tersebut dibahas secara detail berdasarkan penelitian yang selesai dilakukan.

BAB VI : PENUTUP

Dalam bab ini diuraikan mengenai kesimpulan, yaitu menyimpulkan suatu hasil pembahasan dari suatu penelitian yang merupakan hasil akhir dan sekaligus merupakan jawaban dari permasalahan yang ada. Di samping itu juga disertakan saran-saran sebagai arahan dan pendapat yang mungkin dapat bermanfaat bagi peneliti

yang sejenis dalam mengembangkan penelitian lebih lanjut mengenai aplikasi pengurusan dokumen ini.

3. Bagian Akhir Laporan Tugas Akhir

Bagian akhir memuat daftar pustaka yang digunakan sebagai referensi penelitian baik berupa jurnal, artikel hasil penelitian maupun buku dan lampiran surat kesediaan meringkas, dan hal-hal yang berkaitan dengan penyusunan laporan penelitian yang perlu dilampirkan.

BAB II

TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Ratih Titi Komala Sari yang berjudul “Implementasi Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis A,B,C Menggunakan *Tools Expert System Builder*”. Penelitian ini bermula pada awal tahun 2017 dimana Wilayah Kotamadya Depok khususnya, dikagetkan dengan munculnya endemi Hepatitis yang menyebar dilingkungan sekolah yang disebabkan oleh pola makan siswa yang kurang sehat. Untuk mengetahui gejala fisik seseorang terkena penyakit Hepatitis harus segera dilakukan pemeriksaan diinstalasi kesehatan terdekat, dimana memerlukan biaya dan waktu. Sirosis Hepatitis sendiri adalah penyakit hati menahun yang ditandai dengan pembentukan jaringan ikat disertai nodul. Dimana penderita dengan gejala awal Hepatitis tidak menyadarinya hingga dilakukan pemeriksaan secara teliti dilaboratium sehingga telah terjadi pembentukan luka. Pada penelitian ini, penulis membagi Hepatitis kedalam Hepatitis A (HAV), Hepatitis B (HBV) dan Hepatitis C (HCV). Tujuan penelitian ini adalah mendiagnosis penyakit Hepatitis sebagai langkah awal untuk menerapkan kecerdasan buatan dalam dunia medis, merancang dan mengaplikasikan sistem pakar yang mampu mendiagnosis penyakit Hepatitis. Memanfaatkan kecerdasan buatan yang menerapkan metode penalaran maju (*Forward Chaining*) , *tools* yang digunakan dalam penelitian ini adalah *Expert System Builder*, yang memiliki tiga tahapan dalam membangun sistem pakar, yaitu tahap awal mengisikan data primer, tahap kedua pertanyaan, dan tahap ketiga berupa tampilan akhir untuk *End User* untuk mencoba sistem yang telah dibuat [8].

Sedangkan hasil dari penelitian Akhmad Ramdhani, Rizal Ismanto, dan Ike Pertiwi di jurnalnya yang berjudul “Pengembangan Sistem Pakar Untuk Diagnosis Penyakit Hepatitis Berbasis Web menggunakan Metode *Certainty Factor*”. Perkembangan teknologi informasi dewasa ini sangat pesat, hampir semua bidang terpengaruh oleh perkembangan ini. Salah satunya adalah bidang kesehatan, karna keterbatasan Dokter Spesialis dalam waktu praktik mendorong penulis melakukan penelitian untuk membantu tenaga kesehatan dalam melakukan diagnosis awal

Hepatitis dengan hasil yang lebih akurat untuk selanjutnya dilakukan pemeriksaan laboratorium, sehingga pasien dapat langsung mendapatkan perawatan. Dijelaskan dalam penelitiannya bahwa virus Hepatitis merupakan bagian dari keluarga virus A, B, C, D dan E yang menyebabkan peradangan pada hati dimana faktor yang menyebabkan antara lain: virus, infeksi bakteri, racun, obat-obatan tertentu, alkohol serta penyakit lainnya. Metode yang digunakan pada sistem ini adalah *Certainty Factor*, sistem pakar ini dibuat dalam bentuk *Web* berbasis *PHP* dan basis data *MySQL* sehingga memudahkan pengguna dalam mengaksesnya tanpa perlu mengunduh aplikasi. Berdasarkan hasil dari pengujian yang telah dilakukan maka diperoleh hasil pengujian bahwa dengan metode *Certainty Factor* telah berhasil diterapkan dalam sistem pakar ini dengan keputusan berdasarkan gejala yang dimiliki oleh setiap penderita dengan gejala yang berbeda. Hasil dari pengujian sistem ini mencapai keakuratan 86,76%[3].

Penelitian lain yang relevan dengan judul “Sistem Pakar Mendiagnosis Penyakit Hepatitis Menggunakan Metode *Certainty Factor* Berbasis *Web*”. Penelitian bermula karena penyakit hepatitis dapat menyerang semua orang, tak terkecuali orang yang memiliki kekebalan tubuh yang baik. Dengan meningkatnya gaya hidup yang tidak sehat, tentu terdapat resiko penyakit yang mengancam kedepannya. Efek buruk dari gaya hidup yang tidak sehat adalah munculnya gejala penyakit hepatitis yang diakibatkan oleh banyaknya racun, obat-obatan tertentu serta konsumsi alkohol yang dapat menimbulkan kerusakan pada hati. Sistem ini dibuat untuk membantu para pasien untuk mendiagnosis penyakit hepatitis dengan gejala-gejala yang ada. Aplikasi ini diharapkan dapat memberikan informasi kepada masyarakat, untuk mengetahui apa itu Hepatitis serta langkah pencegahannya. Sistem ini dibuat menggunakan metode *Certainty Factor*. Di dalam membangun sistem ini dibuat dengan metode pembangunan perangkat lunak *Waterfall* sebagai proses pengembangan perangkat lunak berurutan yang melewati fase perencanaan, pemodelan, implementasi dan pengujian. Aplikasi ini dibuat untuk melakukan diagnosis sekaligus memberikan informasi terhadap penyakit Hepatitis. Diharapkan dengan adanya aplikasi ini mampu membuat pengguna bisa lebih acuh terhadap kesehatannya dimasa sekarang [9].

Ayu Tiffany Novarina, Novarina dan Indriati melakukan penelitian dengan judul “Sistem Pakar Diagnosis Penyakit Hepatitis Menggunakan Metode *Dempster Shafer*”. Permasalahan yang terjadi saat ini yaitu masih banyak orang awam yang kurang peduli terhadap kesehatan, bahkan tidak jarang masyarakat tidak sadar ketika mereka terserang penyakit karna tidak mengetahui gejala-gejala sehingga terlambat pengobatannya. Pada penelitian ini, penulis menyelesaikan masalah tersebut dengan membuat sistem pakar yang mengimplementasikan metode *Dempester Shafer* sehingga diharapkan dapat mempermudah, mempercepat dan menghemat biaya diagnosis serta operasional [10].

Penelitian dalam Jurnal Riset Komputer (JURIKOM), oleh Elvira Frisca Nasution yang berjudul “Rancangan Aplikasi Sistem Pakar Diagnosis Penyakit Pankreatitis Dengan Metode *Certainty Factor*”. Latar belakang dilakukan penelitian ini bermula karena penyakit dalam yang banyak di derita oleh masyarakat adalah pankreatitis. Penyakit ini menyebabkan inflamasi yang menimbulkan nyeri dimana enzim pankreas diaktifasi secara prematur yang menyebabkan *autodigestif* pada pankreas Pankreatitis sendiri adalah penyakit inflamasi atau peradangan pada pankreas yang umumnya disebabkan oleh minuman beralkohol.. Sistem ini memanfaatkan *Certainty Factor* sebagai metode penalaranya. Dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 2008*, sistem ini berbasis aplikasi *desktop* sehingga memudahkan tenaga kesehatan dalam melakukan proses diagnosis. Sistem ini menyediakan beberapa informasi yang berhubungan dengan pankreas, antara lain: bagaimana perlakuan pankreas, pantangan makanan, saran untuk orang dengan penyakit pankreas, dan rekomendasi pengobatan pankreatitis [11].

Dari tinjauan yang sudah ada, GAP dari sistem yang akan dibuat dengan sistem yang sudah ada sebelumnya adalah sebagai berikut:

Tabel 2.1 GAP dari sistem yang akan dibuat.

No.	Penelitian/Judul	Kesimpulan	Pembeda
1.	Implementasi Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis A,B,C Menggunakan <i>Tools Expert System Builder</i>	<ul style="list-style-type: none"> - Studi kasus yang dibahas adalah penyakit hepatitis khususnya di Kota Depok dengan jenis hepatitis A, hepatitis B, dan hepatitis C. - <i>Tools</i> yang digunakan dalam penelitian ini adalah <i>Expert System Builder</i> dan berbasis aplikasi desktop. - Menggunakan <i>Forward Chaining</i> dan <i>Backward Chaining</i> sebagai metode penalaranya. 	<ul style="list-style-type: none"> - Sistem pakar berbasis <i>Web</i> serta menggunakan PHP sebagai bahasa pemrogramanya. - Menggunakan <i>Forward Chaining</i> dan <i>Certainty Factor</i> sebagai metode penalaranya.
2.	Pengembangan Sistem Pakar Untuk Diagnosis Penyakit Hepatitis Berbasis Web menggunakan Metode <i>Certainty Factor</i>	<ul style="list-style-type: none"> - Studi Kasus yang dibahas adalah penyakit hepatitis dengan jenis hepatitis A, B, C, D, dan E. - Menggunakan <i>Certainty Factor</i> 	<ul style="list-style-type: none"> - Studi kasus yang dibahas adalah penyakit hepatitis dengan jenis hepatitis A, hepatitis B, dan hepatitis C. - Sistem pakar ini dibuat menggunakan

		<p>sebagai metode penalaranya.</p> <ul style="list-style-type: none"> - Sistem pakar ini dibuat dalam bentuk <i>Web</i> dengan PHP sebagai bahasa pemrogramannya dan basis data <i>MySQL</i> 	<p><i>Framework Codeigniter</i> dan basis data <i>MySQLi</i>.</p>
3.	<p>Sistem Pakar Mendiagnosis Penyakit Hepatitis Menggunakan Metode <i>Certainty Factor</i> Berbasis <i>Web</i></p>	<ul style="list-style-type: none"> - Studi kasus yang dibahas adalah penyakit hepatitis dengan jenis hepatitis A, hepatitis B, dan hepatitis C. Serta sudah terinfeksi secara akut - Sistem pakar ini berbasis <i>Web</i>. - Metode yang digunakan adalah <i>Certainty Factor</i>. - Menggunakan metode <i>Waterfall</i> sebagai proses pengembangan perangkat lunak. 	<ul style="list-style-type: none"> - Studi kasus yang dibahas adalah penyakit hepatitis dengan diagnosa awal sesuai dengan gejala yang dirasakan <i>user</i>. - Menggunakan metode pengembangan sistem <i>SDLC Waterfall</i>.
4.	<p>Sistem Pakar Diagnosis Penyakit Hepatitis</p>	<ul style="list-style-type: none"> - Studi Kasus yang dibahas adalah penyakit hepatitis 	<ul style="list-style-type: none"> - Studi kasus yang dibahas adalah penyakit hepatitis

	Menggunakan Metode <i>Dempster Shafer</i>	dengan jenis hepatitis A, B, C, D, dan E. - Penulis menggunakan <i>Dempster Shafer</i> sebagai metode penalaranya.	dengan jenis hepatitis A, hepatitis B, dan hepatitis C. - Menggunakan <i>Forward Chaining</i> dan <i>Certainty Factor</i> sebagai metode penalaranya.
5.	Rancangan Aplikasi Sistem Pakar Diagnosis Penyakit Pankreatitis Dengan Metode <i>Certainty Factor</i>	- Studi Kasus yang dibahas adalah penyakit pankreatitis pada pankreas. - Metode <i>Certainty Factor</i> diterapkan dalam sistem pakar untuk mendiagnosa penyakit pankreatitis dimana hasil dalam bentuk persentase. - Sistem pakar ini dikembangkan menggunakan Visual Basic 2008 dan berbasis <i>Desktop</i> .	- Studi kasus yang dibahas adalah penyakit hepatitis dengan jenis hepatitis A, hepatitis B, dan hepatitis C. - Sistem pakar dikembangkan menggunakan bahasa pemrograman PHP, dan berbasis <i>Web</i> .

BAB III

LANDASAN TEORI

3.1 Sistem Pakar

3.1.1 Pengertian Sistem Pakar

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan. Sistem pakar dirancang agar dapat melakukan penalaran seperti layaknya seorang pakar pada suatu bidang keahlian tertentu.

Melalui sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang hanya dapat diselesaikan dengan bantuan para ahli. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh banyak orang [11].

3.1.2 Kelebihan Dan Kekurangan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti [11]:

- a. Meningkatkan ketersediaan.
- b. Mengurangi biaya (*reduced cost*), biaya yang diperlukan untuk menyediakan keahlian satu orang pengguna menjadi berkurang.
- c. Mengurangi bahaya (*reduced danger*), sistem pakar dapat digunakan dilingkungan yang mungkin berbahaya bagi manusia.
- d. Permanen (*permanence*), sistem pakar dan pengetahuan yang terdapat didalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.
- e. Keahlian multiple (*multiple expertise*), pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinu menyelesaikan suatu masalah setiap saat. Tingkat

keahlian/pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar

- f. Meningkatkan kehandalan (*increased reliability*), sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku, jika sistem dibuat oleh salah seorang pakar sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan yang mungkin terjadi pada saat tertekan atau stress.
- g. Penjelasan (*explanation*), sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah atau tidak mampu melakukannya setiap waktu. Dan hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.
- h. Respon yang cepat (*fast respons*), respon yang cepat/real-time diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.
- i. Basis data cerdas (*intelligent database*), sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

Di samping memiliki beberapa kelebihan dan manfaat, sistem pakar juga memiliki beberapa kelemahan antara lain [11]:

- a. Biaya yang diperlukan untuk membuat dan memeliharanya mahal.
- b. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar dibidangnya.
- c. Sistem pakar tidak 100% bernilai benar.

3.2 Metode Forward Chaining

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF –THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru

(bagian THEN) ditambahkan ke dalam database Langkah-langkah dalam membuat sistem pakar dengan menggunakan metode *forward chaining* yaitu [6]:

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

Sebagai contoh, berikut ini adalah sebuah contoh aturan yang diberikan oleh seorang pakar:

1. Menentukan Tabel Penyakit

Tabel 3.1 Tabel Penyakit

No	Kode Penyakit	Nama Penyakit
1	P1	Hepatitis A
2	P2	Hepatitis B
3	P3	Hepatitis C

2. Menentukan Tabel Keputusan

Tabel 3.2 Tabel Keputusan

No	Kode	Gejala Penyakit	P1	P2	P3
1	G1	Demam ringan	V	V	V
2	G2	Lemah	V	V	V
3	G3	Mual	V	V	V
4	G4	Sakit otot	V	V	V
5	G5	Sakit kepala	V	V	V

6	G6	Diare			V
7	G7	Nyeri sendi	V	V	
8	G8	Sakit perut	V	V	
9	G9	Ikterus	V	V	
10	G10	Tidak nafsu makan	V		

3. Maka dapat ditentukan Tabel Pembentukan Rule

Tabel 3.3 Tabel Pembentukan Rule

RULE	IF	THEN
R1	G1, G2, G3, G4, G5, G7, G8, G9, G10	P1
R2	G1, G2, G3, G4, G5, G7, G8, G9	P2
R3	G1, G2, G3, G4, G5, G6	P3

3.3 Metode *Certainty Factor* (Faktor Kepastian)

Certainty Factor merupakan suatu metode yang digunakan untuk menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. Secara konsep, *Certainty Factor* (CF) merupakan salah satu teknik yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan. *Certainty Factor* (CF) dapat terjadi dengan berbagai kondisi. Diantara kondisi yang terjadi adalah terdapat beberapa antensenden (dalam rule yang berbeda) dengan satu konsekuen yang sama. Dalam kasus ini, kita harus mengagregasikan nilai CF keseluruhan dari setiap kondisi yang ada. Pada konsep *Certainty Factor* ini juga sering dikenal dengan adanya *believe* dan *disbelieve*. *Believe* merupakan keyakinan, sedangkan *Disbelieve* merupakan ketidakpercayaan [5].

Rule - rule yang baru tersebut kemudian dihitung nilai CF pakar dengan CF user menggunakan persamaan:

$$CF [H,E]= MB[H,E] - MD[H,E]$$

Keterangan:

$CF [H,E]$: *certainty factor* hipotesa yang dipengaruhi oleh evidence diketahui dengan pasti. Faktor kepastian dari hipotesis H yang dipengaruhi oleh gejala E. Besarnya CF antara -1 sampai dengan 1. Nilai -1 menunjukkan

ketidakpercayaan mutlak, sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB [H,E]$: *measure of belief terhadap hipotesa* , jika diberikan evidence E (antara 0 dan 1). Ukuran kenaikan kepercayaan terhadap hipotesis H yang dipengaruhi oleh gejala E.

MD : *Measure of Disbelief* (Nilai Ketidakpercayaan)

1. Kelebihan *Certainty Factor*

- a. Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosis penyakit sebagai salah satu contohnya.
- b. Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengelola dua data saja sehingga keakuratan data dapat terjaga [10].

2. Kekurangan *Certainty Factor*

- a. Ide umum dari pemodelan ketidakpastian manusia dengan menggunakan numerik metode *certainty factor* biasanya diperdebatkan. Sebagian orang akan membantah pendapat bahwa formula untuk metode *certainty factor* diatas memiliki sedikit kebenaran.
- b. Metode ini hanya dapat mengolah ketidakpastian/kepastian hanya dua data saja. Perlu dilakukan beberapa kali pengolahan data untuk data yang lebih dari dua buah.
- c. Nilai CF yang diberikan bersifat subyektif karena penilaian setiap pakar bisa saja berbeda-beda tergantung pengetahuan dan pengalaman pakar .

Berikut contoh perhitungan nilai kepastian *Certainty Factor* berdasarkan gejala serta penyakit yang telah disebutkan sebelumnya. Perhitungan dilakukan dengan mengolah input dari pakar dengan memasukan bobot gejala (1-100%), kemudian dilakukan penentuan rule, CF kombinasi masing-masing rules, CF gabungan dilanjutkan pengambilan keputusan.

- a. Pada contoh berikut menggunakan *rule* nomor 3, pengguna memilih enam gejala dengan masing-masing nilai bobot sebagai berikut: [12]

Tabel 3.4 Tabel Input Pakar

Kode gejala	Nama gejala	Input pakar (%)
G1	Demam ringan	36
G2	Lemah	34
G3	Mual	29
G4	Sakit otot	23
G5	Sakit kepala	31
G6	Diare	35

- b. Selanjutnya menentukan bobot CF dari *user*

Tabel 3.5 Tabel Input User

Input user	Keterangan
0	Tidak
0,5	Sedikit yakin
1	Yakin

- c. CF kombinasi didapatkan dengan mengalikan CF *user* dan CF pakar. Sedangkan untuk gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0

Tabel 3.6 Tabel Nilai CF Kombinasi

Kode gejala	Nama gejala	CF user	CF pakar	CF kombinasi
G1	Demam ringan	1	0,36	0,36
G2	Lemah	1	0,34	0,34
G3	Mual	0,5	0,29	0,145
G4	Sakit otot	0,5	0,23	0,115
G5	Sakit kepala	1	0,31	0,31
G6	Diare	1	0,35	0,35

- d. Berdasarkan nilai CF kombinasi pada tabel 3.6 ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan:

$$\text{CFgabungan (CF1, CF2)} = \text{CF1} + \text{CF2} * (1 - \text{CF1})$$

1. $\text{CFgabungan (G1, G2)} = 0,36 + 0,34 * (1 - 0,36)$
 $= 0,36 + 0,2176$
 $= \mathbf{0,5776}$
2. $\text{CFgabungan (old, G3)} = 0,5776 + 0,145 * (1 - 0,5776)$
 $= 0,5776 + 0,061248$
 $= \mathbf{0,638848}$
3. $\text{CFgabungan (old, G4)} = 0,638848 + 0,115 * (1 - 0,638848)$
 $= 0,638848 + 0,0415324$
 $= \mathbf{0,68038048}$
4. $\text{CFgabungan (old, G5)} = 0,68038048 + 0,31 * (1 - 0,68038048)$
 $= 0,68038048 + 0,0990820512$
 $= \mathbf{0,7794625312}$
5. $\text{CFgabungan (old, G6)} = 0,7794625312 + 0,35 * (1 - 0,7794625312)$
 $= 0,7794625312 + 0,0771881141$
 $= \mathbf{0,8566506453}$

- e. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $\text{Cfgabungan} * 100\%$. Sehingga dihasilkan perhitungan sebesar **85,6%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis C.

3.4 Definisi Hepatitis

Istilah Hepatitis dipakai untuk semua jenis peradangan pada sel-sel hati, yang bisa disebabkan oleh infeksi (virus, bakteri, parasite), obat-obatan (termasuk obat tradisional), konsumsi alkohol, lemak yang berlebih dan penyakit autoimmune. Ada 5 jenis Hepatitis Virus yaitu Hepatitis A, B, C, D, dan E. Antara Hepatitis yang satu dengan yang lain tidak saling berhubungan.

Pada penelitian ini hanya dibahas mengenai penyakit Hepatitis A, B, dan C yang sering ditemui dikarenakan Virus Hepatitis D dan E jarang ditemui [12].

Tabel 3.7 Prevalensi Hepatitis di Indonesia Tahun 2019

Karakteristik	Prevalensi Hepatitis (%)
Kelompok umur (tahun)	
<1	0,5
1 - 4	0,8
5 - 14	1,0
15 - 24	1,1
25 - 34	1,3
35 - 44	1,3
45 - 54	1,4
55 - 64	1,3
65 - 74	1,4
>75	1,3
Jenis Kelamin	
Laki-laki	1,3
Perempuan	1,1
Pekerjaan	
Tidak Bekerja	1,1
Pegawai	1,0
Wiraswasta	1,2
Petani/Nelayan/Buruh	1,6
	1,4

Dari table di atas terlihat karakteristik prevalensi Hepatitis tertinggi pada kelompok umur 45-54 dan 65-74 (1,4%). Penderita Hepatitis baik pada laki-laki maupun perempuan, proporsinya tidak berbeda secara bermakna. Jenis pekerjaan juga mempengaruhi prevalensi Hepatitis, penderita Hepatitis banyak ditemukan pada petani atau nelayan serta buruh dibandingkan jenis pekerjaan yang lain[12] .

3.4.1 Hepatitis A

WHO memperkirakan di dunia setiap tahunnya ada sekitar 1,4 juta penderita Hepatitis A. Di Amerika insidens Hepatitis A adalah 1 per 100.000 penduduk, dengan estimasi 21.000 orang (Tahun 2009). Di Eropa insidens Hepatitis A adalah 3,9 per 100.000 penduduk (Publikasi tahun 2008). Di Indonesia, Hepatitis A sering muncul dalam Kejadian Luar Indonesia (KLB). Tahun 2010 tercatat 6 KLB dengan jumlah penderita 279, jumlah kematian 0, CFR 0 sedangkan tahun 2011 tercatat 9 KLB, jumlah penderita 550, jumlah kematian 0,

CFR 0. Tahun 2012 sampai bulan Juni, telah terjadi 4 KLB dengan jumlah penderita 204, jumlah kematian 0, CFR 0 [13].

a. Etiologi

Penyebab penyakit adalah virus Hepatitis A (VHA), termasuk family picornaviridae berukuran 27 nanometer, genus hepatovirus yang dikenal sebagai enterovirus 72, mempunyai 1 serotype dan 4 genotype, merupakan RNA virus. Virus Hepatitis A bersifat termotabil, tahan asam dan tahan terhadap empedu. Virus ini diketahui dapat bertahan hidup dalam suhu ruangan selama lebih dari 1 bulan. Pejamu infeksi VHA hanya terbatas pada manusia dan beberapa binatang primata. Virus dapat diperbanyak secara in vitro dalam kultur sel primer monyet kecil atau secara invivo pada simpanse [13].

b. Cara Penularan

Menurut Direktorat Jenderal Kementerian Kesehatan RI pada tahun 2015 [13], virus Hepatitis A ditularkan secara fecal-oral. Virus ini masuk ke dalam saluran pencernaan melalui makanan dan minuman yang tercemar tinja penderita VHA. Virus kemudian masuk ke hati melalui peredaran darah untuk selanjutnya menginvasi sel-sel hati (hepatosit), dan melakukan replikasi dihepatosit. Jumlah virus yang tinggi dapat ditemukan dalam tinja penderita sejak 3 hari sebelum muncul gejala hingga 1-2 minggu setelah munculnya gejala kuning pada penderita. Ekskresi virus melalui tinja pernah dilaporkan mencapai 6 bulan pada bayi dan anak. Sebagian besar kasus kemungkinan tidak menular lagi pada minggu pertama setelah ikterus. Ekskresi kronis pada VHA tidak pernah dilaporkan.

Infeksi Hepatitis A sering terjadi dalam bentuk Kejadian Luar biasa (KLB) dengan pola *common source*, umumnya sumber penularan berasal dari air minum yang tercemar, makanan yang tidak dimasak, makanan yang tercemar, dan sanitasi yang buruk.

Selain itu, walaupun bukan merupakan cara penularan yang utama, penularan melalui transfusi atau penggunaan jarum suntik bekas penderita dalam masa inkubasi juga pernah dilaporkan [13].

3.4.2 Hepatitis B

Menurut hasil Riset Kesehatan Dasar pada tahun 2018 [1], menunjukkan bahwa Hepatitis klinis terdeteksi di seluruh propinsi di Indonesia dengan prevalensi sebesar 0,6% (rentang: 0,2%-1,9%). Hasil Riskesdas Biomedis tahun 2007 dengan jumlah sampel 10.391 menunjukkan bahwa persentase HBsAg positif 9,4%. Persentase Hepatitis B tertinggi pada kelompok umur 45-49 tahun (11,92%), umur >60 tahun (10,57%) dan umur 10-14 tahun (10,02%). HBsAg positif pada kelompok laki- laki dan perempuan hampir sama (9,7% dan 9,3%). Hal ini menunjukkan bahwa 1 dari 10 penduduk Indonesia telah terinfeksi virus Hepatitis B.

a. Etiologi

Penyebab penyakit adalah virus Hepatitis B (VHB) yang termasuk family Hepadnavirus dan berukuran sangat kecil (42 nm). Virus Hepatitis B merupakan virus DNA dan sampai saat ini terdapat 8 genotip VHB yang telah teridentifikasi, yaitu genotip A–H. VHB memiliki 3 jenis morfologi dan mampu mengkode 4 jenis antigen, yaitu HBsAg, HBeAg, HBcAg, dan HBxAg. Virus Hepatitis B yang menginfeksi manusia bisa juga menginfeksi simpanse. Virus dari Hepadnavirus bisa juga ditemukan pada bebek, marmut dan tupai tanah, namun virus tersebut tidak bisa menginfeksi manusia [13].

b. Cara Penularan

Menurut Direktorat Jenderal Kementerian Kesehatan RI pada tahun 2015 [13], virus Hepatitis B dapat ditemukan pada cairan tubuh penderita seperti darah dan produk darah, air liur, cairan serebrospinalis, peritonea, pleural, cairan amniotik, semen, cairan vagina dan cairan tubuh lainnya. Namun tidak semuanya memiliki

kadar virus yang infeksius. Secara umum, penularan bisa terjadi secara vertikal maupun horizontal. Untuk saat ini, penularan VHB yang utama diduga berasal dari hubungan intim dan transmisi perinatal. Transmisi horizontal adalah penularan dari satu individu ke individu lainnya. Selain lewat hubungan seksual tidak aman, transmisi horizontal Hepatitis B juga bisa terjadi lewat penggunaan jarum suntik bekas penderita Hepatitis B, transfusi darah yang terkontaminasi virus Hepatitis B, pembuatan tato, penggunaan pisau cukur, sikat gigi, dan gunting kuku bekas penderita Hepatitis B. Sementara itu, berpelukan, berjabat tangan, atau berciuman dengan penderita Hepatitis B belum terbukti mampu menularkan virus ini.

Penularan secara vertikal adalah penularan yang terjadi pada masa perinatal yaitu penularan dari ibu kepada anaknya yang baru lahir, jika seorang ibu hamil karier Hepatitis B dan HBeAg positif maka bayi yang di lahirkan 90% kemungkinan akan terinfeksi dan menjadi karier juga. Kemungkinan 25% dari jumlah tersebut akan meninggal karena Hepatitis kronik atau kanker hati. Transmisi perinatal ini terutama banyak terjadi di negara-negara Timur dan negara berkembang. Infeksi perinatal paling tinggi terjadi selama proses persalinan dan diduga tidak berhubungan dengan proses menyusui [13].

3.4.3 Hepatitis C

Berdasarkan hasil Surveilans Hepatitis C oleh Direktorat Jenderal Pengendalian Penyakit dan Penyehatan Lingkungan pada tahun 2010-2011 yang dilaksanakan di 21 propinsi, 53 rumah sakit, 49 laboratorium dan 26 Unit Transfusi Darah (UTD) PMI, dengan jumlah 1.825.823 sampel, kasus positif 29.480 orang, jumlah kasus terbanyak didapatkan pada golongan umur 20-40 tahun sebanyak 58,5% sedangkan proporsi menurut jenis kelamin menunjukkan bahwa pada kelompok laki-laki 83% dan 17% pada perempuan.

Prevalensi anti-VHC pada beberapa donor di Indonesia berkisar 0,5-3,4%. Prevalensi Anti-VHC pada virus Hepatitis Akut 9,5-20%, prevalensi Anti-VHC pada sirosis hati berkisar 30,8-89,2 persen.

Data ko-infeksi diperoleh dari beberapa penelitian, Rino S Gani (FK-UI, RSCM) penderita dengan HIV (IVDU), ko infeksi 80%, penderita ko infeksi dengan Hepatitis B berkisar 10-19%, Suryanto Sidik (RS Mintoarjo) pada penderita dengan HIV, 31,6% ko-infeksi dengan VHC.

a. Etiologi

Penyebab penyakit Hepatitis C adalah virus Hepatitis C (VHC) yang termasuk family Flaviviridea genus Hepacivirus dan merupakan virus RNA. Setidaknya 6 genotip dan lebih dari 50 subtype VHC yang berbeda telah ditemukan [13].

b. Cara Penularan

Menurut Direktorat Jenderal Kementerian Kesehatan RI pada tahun 2015 [13], cara penularan VHC yang paling umum adalah secara parenteral, yaitu berkaitan dengan penggunaan bersama jarum suntik yang tidak steril terutama pada pengguna obat-obatan terlarang, tato, tindik, penggunaan alat pribadi seperti pisau cukur, sikat gigi bersama penderita, transfusi darah, operasi, transplantasi organ, dan melalui hubungan seksual. VHC adalah penyebab utama dari Hepatitis yang diderita setelah transfusi darah. Walaupun begitu, peraturan yang memperketat pemeriksaan darah bagi donor darah telah menurunkan risiko infeksi secara drastis. Penularan dapat terjadi dalam waktu 1 minggu atau lebih setelah timbulnya gejala klinis yang pertama pada penderita.

3.5 MySQL

MySQL adalah sistem manajemen *database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management sistem (DBMS)*. *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan [14].

MySQL memiliki beberapa kelebihan dan keuntungan dibanding *database* lain, di antaranya adalah [15] :

- a. Banyak ahli berpendapat *MySQL* merupakan server tercepat.
- b. *MySQL* merupakan sistem manajemen *database* yang *Open Source*.
- c. *MySQL* mempunyai performa yang tinggi tapi simpel.
- d. *Database MySQL* mengerti *SQL (Structured Query Language)*.
- e. *MySQL* dapat diakses melalui *protocol ODBC (Open Database Connectivity)* buatan *Microsoft*. Ini menyebabkan *MySQL* dapat diakses oleh banyak *software*.
- f. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses *database*.
- g. *Database MySQL* dapat diakses dari semua tempat di *internet* dengan hak akses tertentu.
- h. *MySQL* merupakan *database* yang mampu menyimpan data berkapasitas besar, sampai berukuran *Gigabyte*.
- i. *MySQL* dapat berjalan di berbagai *oprating system* seperti *Linux*, *Windows*, *Solaris* dan lain-lain.

3.6 PHP

PHP atau yang memiliki kepanjangan (*Hypertext Preprocessor*) merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka layout *web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah di-*maintenance* [16].

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya *web server*.

PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada *web server* apache dan sebagai binary yang dapat berjalan sebagai *CGI*. Keunggulan PHP dari sifatnya yang *server-side* tersebut antara lain [16]:

- a. PHP merupakan program *open source* (tidak memerlukan biaya lisensi).
- b. *Multi-platform* (dapat dijalankan pada sistem operasi yang berbeda-beda).
- c. Adanya penggunaan *session* / sesi.
- d. Tidak diperlukan kompatibilitas *browser* atau harus menggunakan browser tertentu, karena server-lah yang akan mengerjakan skrip PHP. Hasil yang dikirimkan kembali ke *browser* umumnya bersifat teks atau gambar saja sehingga pasti dikenal oleh *browser* apapun.
- e. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*.

3.7 Codeigniter

Codeigniter adalah *Framework* yang dikembangkan pada Bahasa Pemrograman PHP dan menggunakan konsep MVC. *Codeigniter* bersifat *Open-Source* yang banyak digunakan oleh para *developer* dalam mengembangkan website yang dinamis. MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *Model*, *View*, dan *Controller* [16].

a. Model

Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi *database*. Seperti mengambil data dari *database*, *input* data dan pengolahan *database* lainnya. Semua instruksi yang berhubung dengan pengolahan *database* diletakkan di dalam *model*.

b. *View*

View merupakan bagian yang menangani halaman *user interface*. Tampilan ini di kumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan pengguna dalam melakukan pengembangan tampilan.

c. *Controller*

Controller merupakan kumpulan instruksi aksi yang menghubungkan *model* dan *view*, jadi pengguna tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah instruksi.

3.8 Unified Modeling Language (UML)

Secara umum *Unified Modeling Language (UML)* merupakan “bahasa” untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. Dalam kerangka visualisasi, para pengembang menggunakan *UML* sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem/perangkat lunak. Dengan adanya “bahasa” yang bersifat standar, komunikasi perancang dengan pemrogram (lebih tepat lagi komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus [16]. Beberapa literatur menyebutkan bahwa *UML* menyediakan 9 jenis diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi [18].



Jenis diagram-diagram tersebut adalah diagram kelas (*class diagram*), diagram paket (*package diagram*), diagram *use-case* (*use case diagram*), diagram interaksi dan *sequence* (*sequence diagram*), diagram komunikasi (*communication diagram*), diagram komponen (*component diagram*), diagram *deployment* (*deployment diagram*) [19]. Berikut ini adalah penjelasan dari beberapa jenis diagram yang akan digunakan [20] :




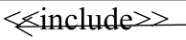
a. *Use case* Diagram

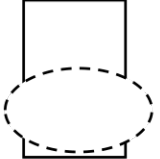

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case* Diagram menampilkan actor mana yang menggunakan *use case*, *use case* mana yang memasukkan *use case* lain dan hubungan antara *actor* dan *use case*.

Use Case Diagram menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Tabel 3.8 berikut merupakan penjelasan *use case* diagram.

Tabel 3.8 Tabel Use Case Diagram

Nama	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja awal di frase nama <i>Use Case</i> .
Aktor		Orang, proses, atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.

Asosiasi		<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. Asosiasi merupakan hubungan statis antar elemen yang menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.</p>
Ekstensi		<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Misalnya arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
Generalisasi		<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya</p>
<i>Include</i>		<p>Relasi <i>use case</i> tambahan ke sebuah <i>use</i></p>

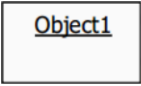



		<i>case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat.
<i>Collaboration</i>		Interaksi aturan-aturan yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
<i>Note</i>		Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
<i>System</i>		Menspesifikasikan paket yang menampilkan sistem secara terbatas.

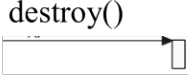
b. *Sequence Diagram*

Sequence diagram merupakan salah satu diagram yang menjelaskan bagaimana suatu operasi dilakukan, di mana perintah apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan berdasarkan waktu terjadinya dalam perintah yang terurut.

Sequence diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah perintah untuk menghasilkan *output* tertentu, dan perubahan apa saja yang terjadi secara internal dan *output* yang dihasilkan. Tabel 3.9 berikut merupakan penjelasan dari *sequence diagram*.

Tabel 3.9 Tabel Sequence Diagram

Nama	Simbol	Keterangan
<i>Actor</i>		Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
Garis hidup		Menyatakan kehidupan suatu objek.
<i>Object</i>		Menyatakan objek yang berinteraksi pesan
Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i>	1:nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i>	1:masukan 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya.
Pesan <i>return</i>	1: keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau suatu metode akan menghasilkan suatu kembalian ke objek tertentu.



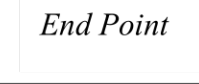


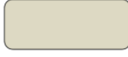


Pesan <i>destroy</i>		Suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destory</i> .
-------------------------	---	---


c. *Activity Diagram*

Activity diagram merupakan teknik untuk mengembangkan logika procedural, proses bisnis, dan jalur kerja. *Activity* diagram sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem atau perangkat lunak yang sedang dikembangkan.

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibantu dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya seperti use case atau interaksi. Tabel 3.10 berikut merupakan penjelasan dari *activity* diagram.

Tabel 3.10 Tabel Activity Diagram

Nama	Simbol	Keterangan
		Menunjukkan dimulainya suatu <i>workflow</i> .
		Menggambarkan akhir pada sebuah <i>activity diagram</i> .
		Menggambarkan pekerjaan atau tugas dalam <i>workflow</i> .
		Titik pada <i>activity diagram</i> yang mengindikasikan suatu kondisi di mana terdapat suatu kemungkinan atau perbedaan transisi.

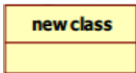




<i>State Transition</i>		Menunjukkan suatu kegiatan berikutnya setelah kegiatan sebelumnya.
-------------------------	---	--



d. *Class Diagram*

Class diagram merupakan jenis diagram struktur statis dalam *UML* yang menggambarkan struktur sistem dengan menunjukkan *class*, atribut, metode, dan hubungan antar objek. *Class* diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan komponen.

Berbagai komponen tersebut dapat mewakili *class* yang akan diprogram, objek utama, atau interaksi antara *class* dan objek. Tabel 3.11 merupakan penjelasan *class diagram*.

Tabel 3.11 Tabel Class Diagram

Nama	Simbol	Keterangan
<i>Class</i>		Kelas pada struktur sistem.
<i>Interface</i>		Kumpulan operasi tanpa implementasi dari suatu <i>class</i> .
<i>Association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
<i>Directed Association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<i>Generalisasi</i>		Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).

<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"><i>Dependency</i></div>		Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"><i>Aggregation</i></div>		Semua bagian (<i>whole-part</i>)

3.9 Black Box Testing

Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan [16].

Tujuan *black box testing* adalah untuk mencari kesalahan/kegagalan dalam operasi, yang mencakup kemampuan dari perangkat lunak, operasional, dan skenario pemakai. Konsep pengujian black box antara lain:

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau *database* eksternal.
- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.

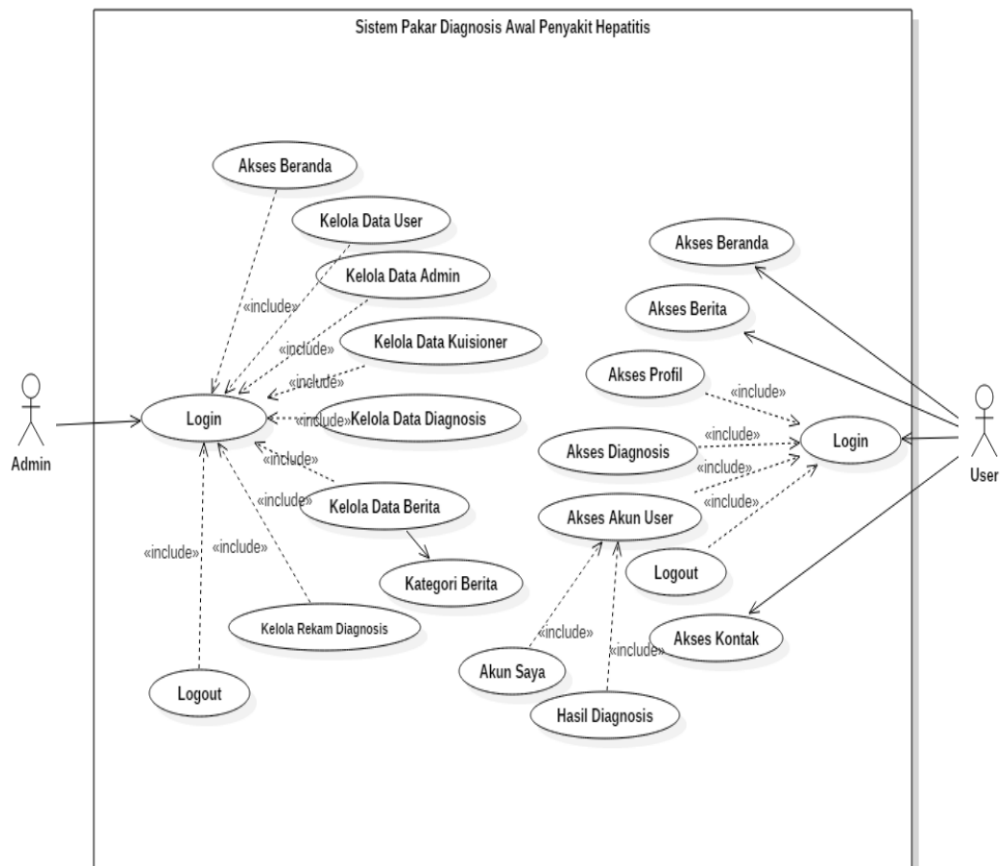
BAB IV DESAIN DAN PERANCANGAN

4.1 Perancangan Sistem

Perancangan sistem merupakan salah satu unsur atau tahapan dari keseluruhan pembuatan sistem, dalam perancangan sistem ini menggunakan *tools UML (Unified Modelling Language)* meliputi *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram*.

4.2 Perancangan *Unified Modelling Language*

4.2.1 Usecase Diagram



Gambar 4.1 Use Case Diagram sistem pakar diagnosis awal hepatitis

Use case diagram menggambarkan hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. Perbedaan antara *Extend* dengan *Include* adalah:

- a. *Extend* adalah relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.
- b. Sedangkan *Include* adalah Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	User	Mempunyai hak akses untuk melakukan konsultasi dan melihat apa saja konsultasi yang telah dilakukan, serta membaca informasi atau berita.
2	Admin	Mempunyai hak akses untuk menambah, merubah dan menghapus data data yang diperlukan.

Tabel 4.2 Identifikasi *use case diagram user*

No	Definisi	Deskripsi
1	Login	Memasukan <i>username</i> dan <i>password</i> yang dibutuhkan untuk melakukan konsultasi.
2	Beranda	Tampilan awal yang tampil ketika user membuka aplikasi, berisi informasi singkat serta berita berita terkait hepatitis.
3	Diagnosis	Proses diagnosis, dimana user akan ditampilkan pertanyaan berupa gejala gejala awal penyakit hepatitis.
4	Profil	Terdapat bagaimana cara menggunakan sistem serta fungsi dibuatnya sistem pakar ini.

5	Kontak	Menampilkan kontak yang dapat dihubungi oleh <i>user</i> , seperti nomor ponsel serta sosial media.
6	<i>User</i>	<i>User</i> dapat mengelola akun mereka seperti merubah data pribadi, serta melihat hasil diagnosis sebelumnya.

Tabel 4.3 Identifikasi use case diagram admin

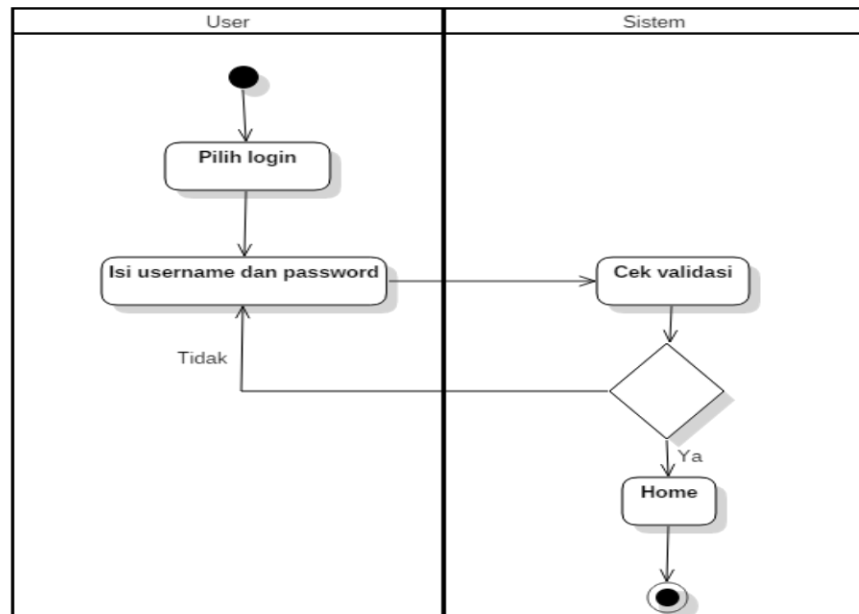
No	Definisi	Deskripsi
1	<i>Login</i>	Memasukan <i>username</i> dan <i>password</i> yang dibutuhkan untuk masuk ke sistem.
2	<i>Dashboard</i>	Berisi informasi singkat mengenai sistem.
3	Data admin	Menampilkan data admin, terdapat fitur tambah, edit serta hapus data admin.
4	Data <i>user</i>	Menampilkan data <i>user</i> , terdapat fitur tambah, edit serta hapus data user.
5	Data kuisisioner	Berisi pertanyaan yang akan diajukan ke user pada proses diagnosis, data dapat ditambah, edit dan hapus.
6	Data diagnosis	Berisi diagnosis penyakit hepatitis hasil dari proses diagnosis, data dapat ditambah, edit dan hapus.
7	Data rekam diagnosis	Berisi hasil diagnosis yang telah dilakukan user, data hanya dapat dihapus.
8	Data berita	Menampilkan informasi atau berita yang berkaitan dengan penyakit hepatitis.

9	Logout	Untuk admin keluar dari sistem
---	--------	--------------------------------

4.2.2 Activity Diagram

1. Activity diagram login (User)

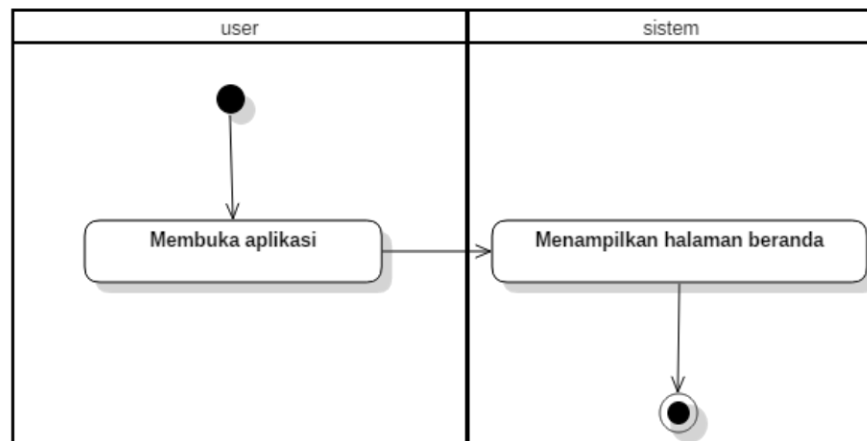
Diagram *login* menjelaskan tentang jalannya aplikasi untuk proses *login*, user memasukan *username* dan *password*. Ketika *username* dan *password* sesuai maka secara otomatis masuk ke tampilan beranda dan jika salah maka akan kembali ke halaman *login*. Adapun contoh *activity diagram login (user)* dapat dilihat pada gambar 4.2



Gambar 4.2 Activity diagram login (User)

2. Activity diagram beranda (User)

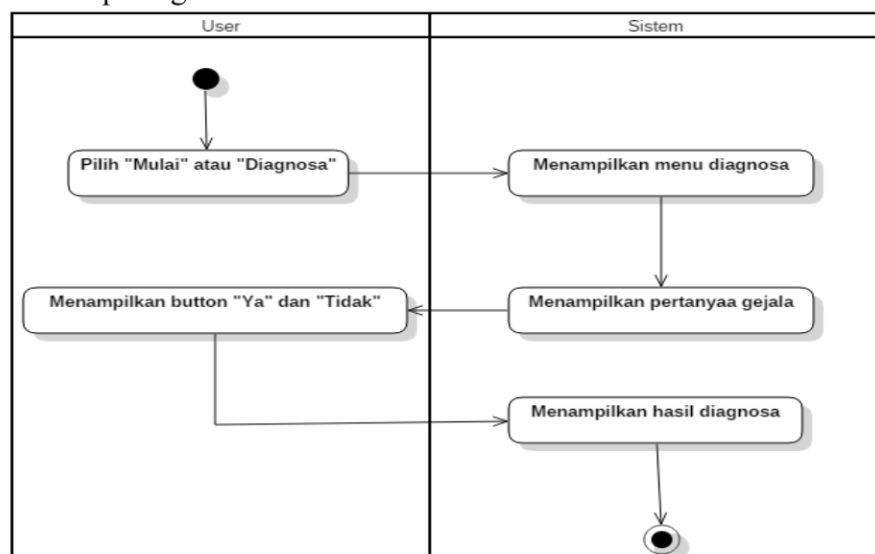
Diagram beranda menjelaskan bagaimana jalanya halaman beranda user, di mana beranda adalah halaman yang pertama muncul saat *user* berhasil *login* dan masuk kedalam aplikasi. Adapun contoh *activity* diagram beranda (*user*) dapat dilihat pada gambar 4.3



Gambar 4.3 Activity diagram beranda (User)

3. Activity diagram diagnosis (User)

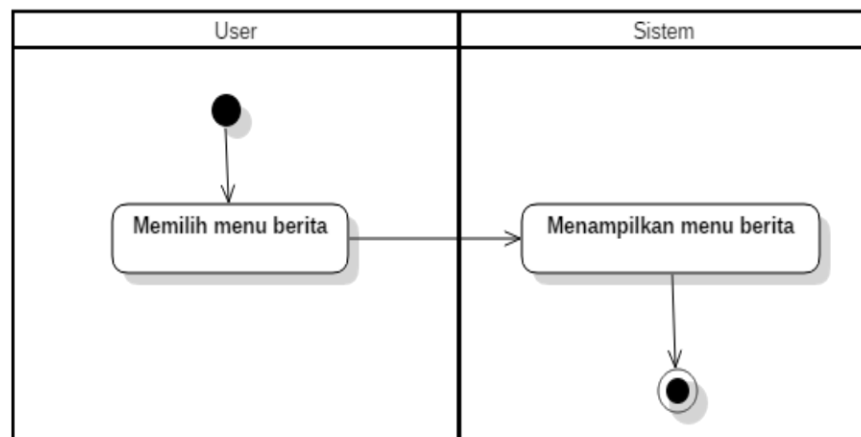
Diagram diagnosis menjelaskan jalanya sistem dimenu diagnosis, di mana user mulai diagnosis berdasarkan gejala yang dirasakan. Adapun contoh *activity* diagram diagnosis (*user*) dapat dilihat pada gambar 4.3



Gambar 4.4 Activity diagram diagnosis (User)

4. *Activity diagram* berita (*User*)

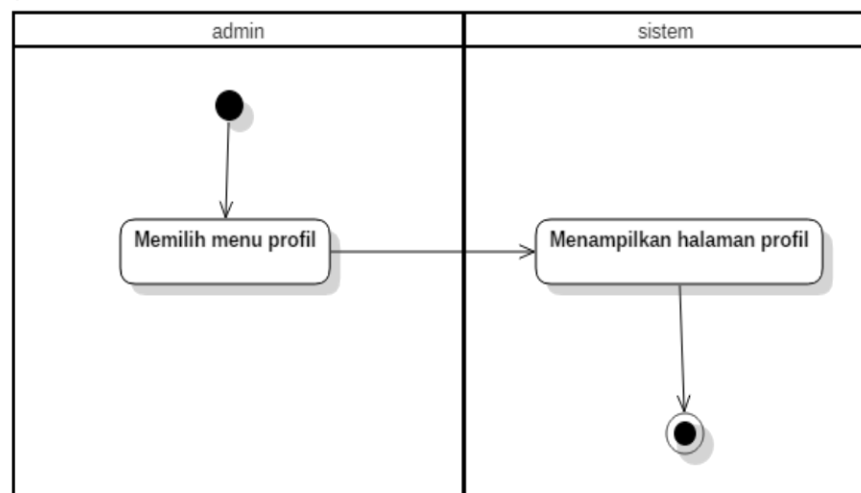
Diagram berita menjelaskan jalanya sistem pada halaman berita, di mana berisi informasi mengenai penyakit hepatitis dan informasi kesehatan lainnya. Adapun contoh *activity* diagram berita (*user*) dapat dilihat pada gambar 4.5



Gambar 4.5 *Activity diagram* berita (*User*)

5. *Activity diagram* profil (*User*)

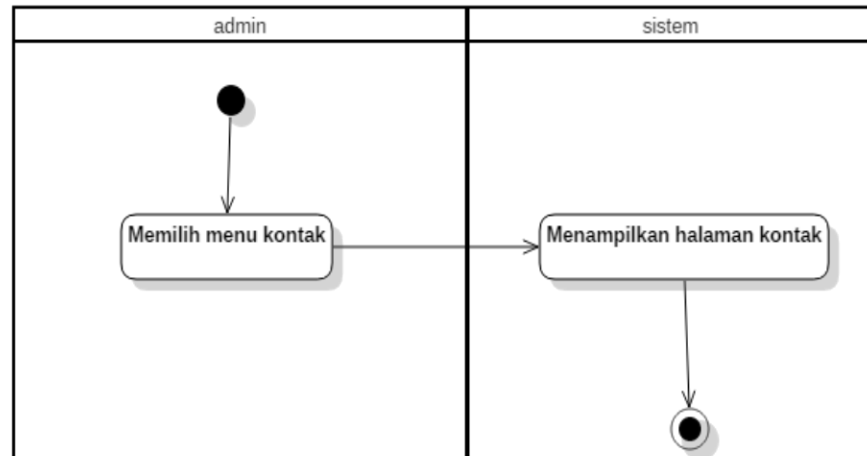
Diagram profil menjelaskan jalanya sistem pada halaman profil, berisi informasi tentang bagaimana cara menggunakan sistem pakar serta terdapat fungsi dibuatnya sistem pakar ini. Adapun contoh *activity* diagram profil (*user*) dapat dilihat pada gambar 4.6



Gambar 4.6 *Activity diagram* profil (*User*)

6. *Activity diagram kontak (User)*

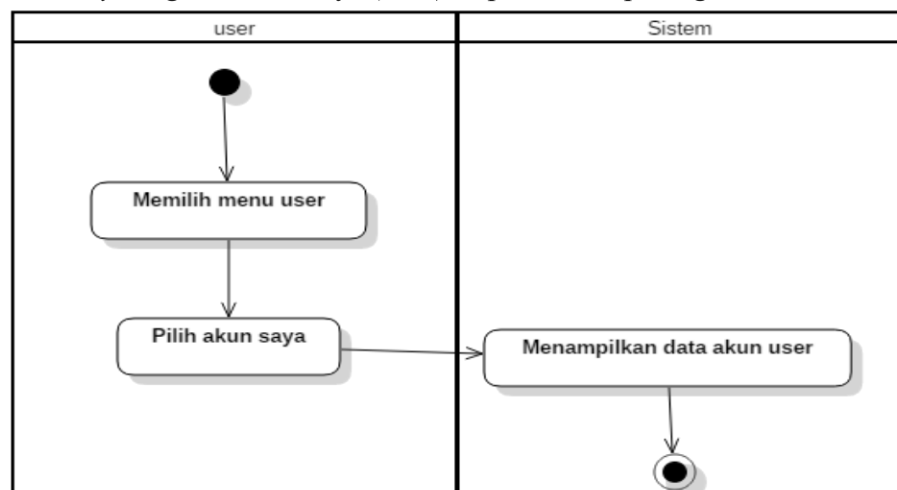
Diagram kontak menjelaskan jalanya sistem pada halaman kontak, di mana terdapat kontak yang dapat dihubungi oleh *user* jika ada pertanyaan atau bantuan saat menggunakan aplikasi. Adapun contoh *activity diagram kontak (user)* dapat dilihat pada gambar 4.7



Gambar 4.7 *Activity diagram kontak (User)*

7. *Activity diagram akun saya (User)*

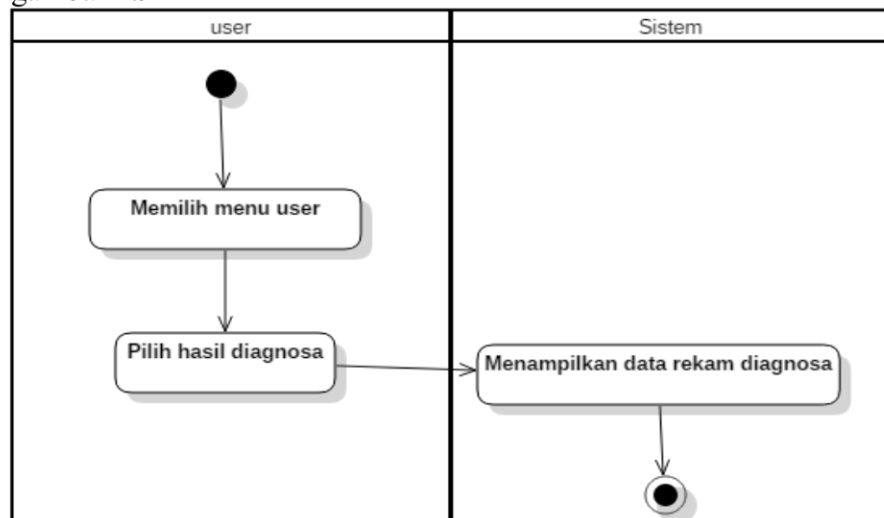
Diagram akun saya menjelaskan jalanya alur sistem ketika *user* akan mengelola data pribadi pada akun miliknya, *user* dapat merubah *password*, nomor hp, dan foto profil. Adapun contoh *activity diagram akun saya (user)* dapat dilihat pada gambar 4.8



Gambar 4.8 *Activity diagram akun saya (User)*

8. Activity diagram hasil diagnosis (User)

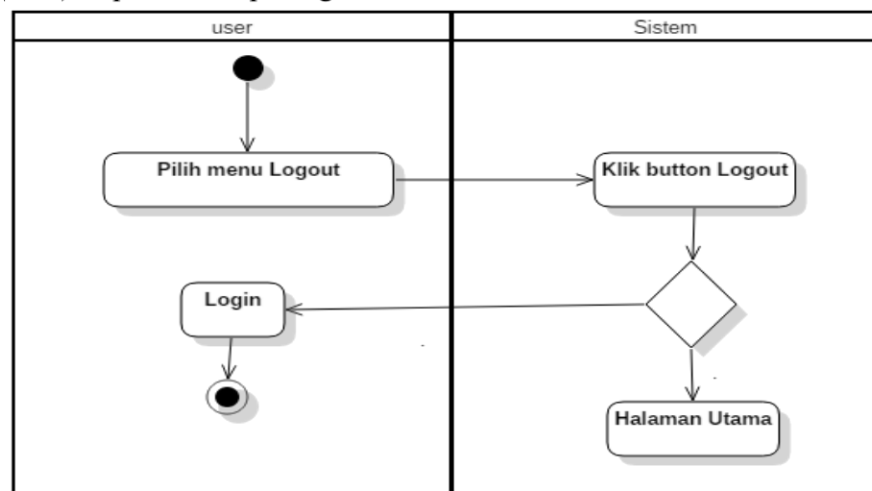
Diagram hasil diagnosis menjelaskan jalanya alur sistem pada halaman hasil diagnosis, rekam diagnosis yang telah dilakukan oleh *user* sebelumnya dapat dilihat detailnya pada halaman ini. Adapun *activity* diagram hasil diagnosis (*user*) dapat dilihat pada gambar 4.9



Gambar 4.9 Activity diagram hasil diagnosis (User)

9. Activity diagram logout (User)

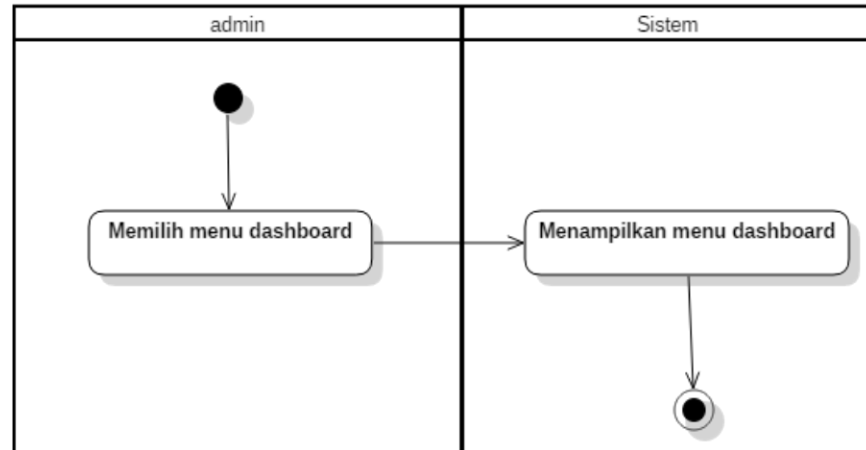
Diagram *logout* menjelaskan alur sistem ketika *user* logout dari sistem, di mana akan kembali ke halaman awal atau *user* dapat login kembali ke dalam sistem. Adapun *activity* diagram logout (*user*) dapat dilihat pada gambar 4.10



Gambar 4.10 Activity diagram logout (User)

10. Activity diagram beranda (Admin)

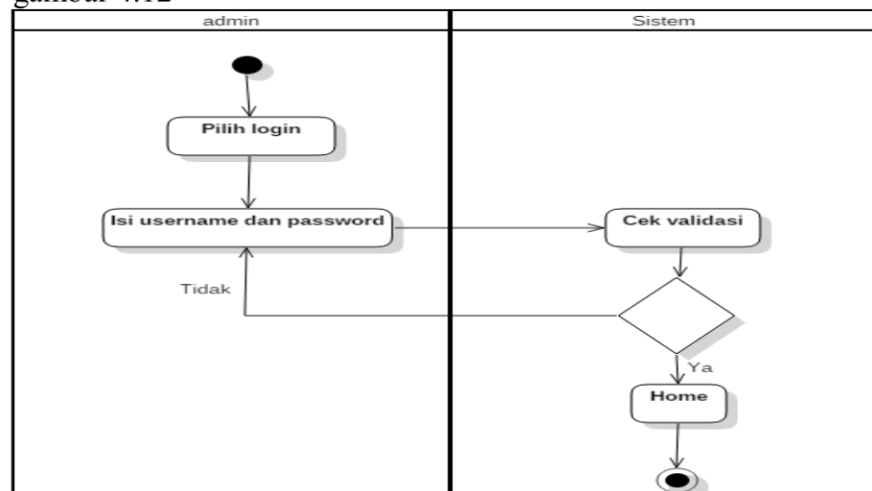
Diagram beranda menjelaskan tentang jalannya aplikasi untuk tampilan beranda (*dashboard*), pada dashboard terdapat informasi singkat tentang sistem seperti *history* diagnosis terbaru, dll. Adapun *activity* diagram beranda (*admin*) dapat dilihat pada gambar 4.11



Gambar 4.11 Activity diagram dashboard (Admin)

11. Activity diagram login (Admin)

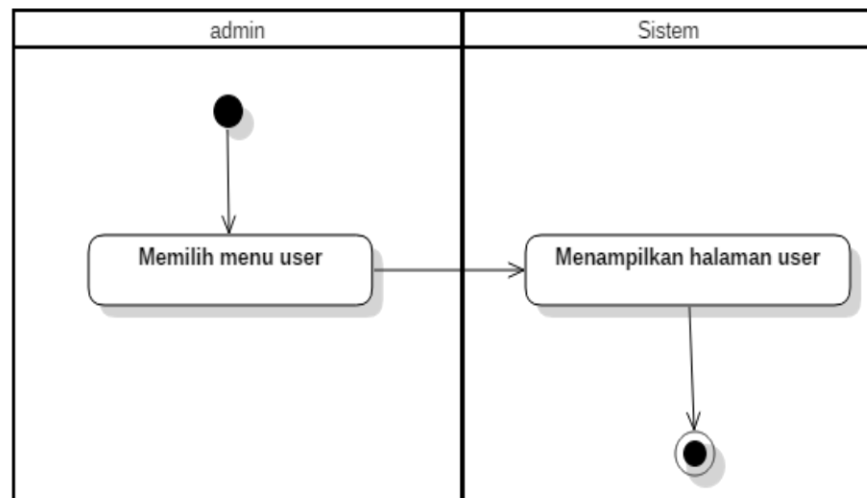
Diagram *login* menjelaskan tentang jalannya aplikasi untuk proses *login*, user memasukkan *username* dan *password*. Ketika *username* dan *password* sesuai maka secara otomatis masuk ke tampilan beranda dan jika salah maka akan kembali ke halaman *login*. Adapun *activity* diagram *login* (*admin*) dapat dilihat pada gambar 4.12



Gambar 4.12 Activity diagram login (Admin)

12. Activity diagram User (Admin)

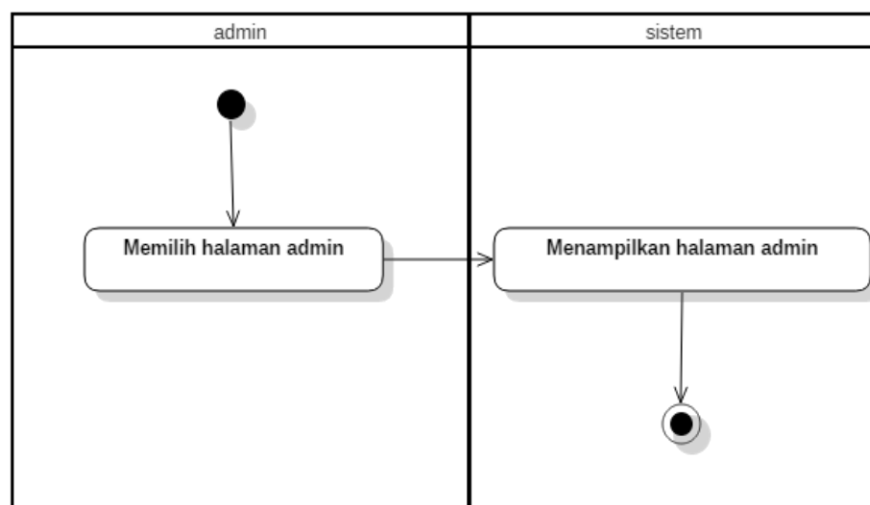
Diagram ini menjelaskan jalanya alur sistem ketika menampilkan halaman *user*. Di mana *admin* dapat mengolah sumber daya *user* mulai dari tambah, edit, dan hapus. Adapun *activity* diagram user (*admin*) dapat dilihat pada gambar 4.13



Gambar 4.13 Activity diagram user (Admin)

13. Activity diagram admin (Admin)

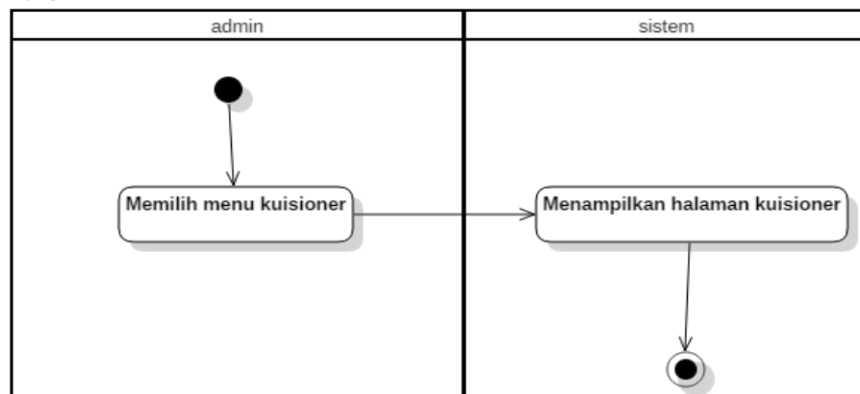
Diagram ini menjelaskan alur sistem ketika mengakses halaman *admin*, di mana *admin* dapat mengelola sumber daya sistem berupa data *admin* yang dapat di tambah, edit, dan hapus. Adapun *activity* diagram *admin* (*admin*) dapat dilihat pada gambar



Gambar 4.14 Activity diagram admin (Admin)

14. Activity diagram gejala (Admin)

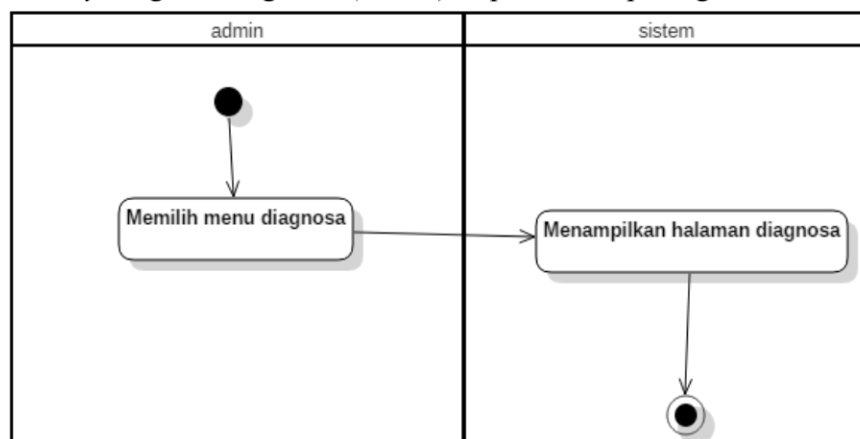
Diagram gejala menjelaskan jalanya alur sistem ketika *admin* mengakses halaman gejala. Pada halaman ini, terdapat daftar pertanyaan yang bersangkutan dengan gejala penyakit hepatitis. *Admin* dapat melakukan tambah, edit, serta hapus pada data gejala. Adapun *activity* diagram gejala (*admin*) dapat dilihat pada gambar 4.15



Gambar 4.15 Activity diagram gejala (Admin)

15. Activity diagram diagnosis (Admin)

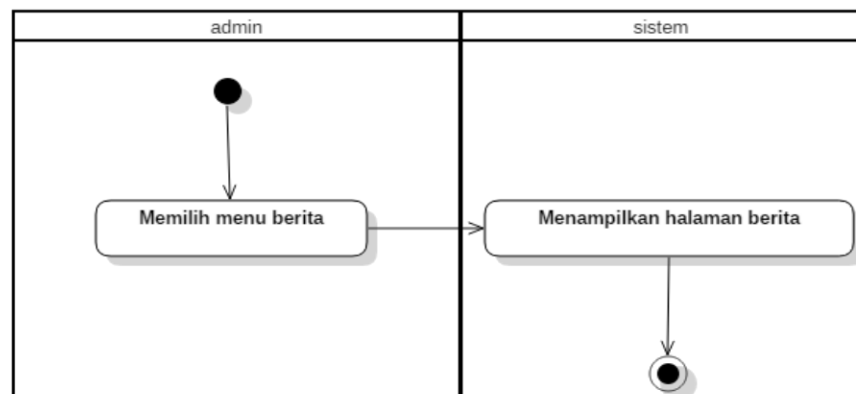
Diagram diagnosis menjelaskan alur sistem ketika *admin* mengakses halaman diagnosis. *Admin* dapat mengelola sumber daya data diagnosis yang berisi jenis penyakit hepatitis serta dapat melakukan olah data seperti tambah, edit, dan hapus. Adapun *activity* diagram diagnosis (*admin*) dapat dilihat pada gambar 4.16



Gambar 4.16 Activity diagram diagnosis (Admin)

16. Activity diagram berita (Admin)

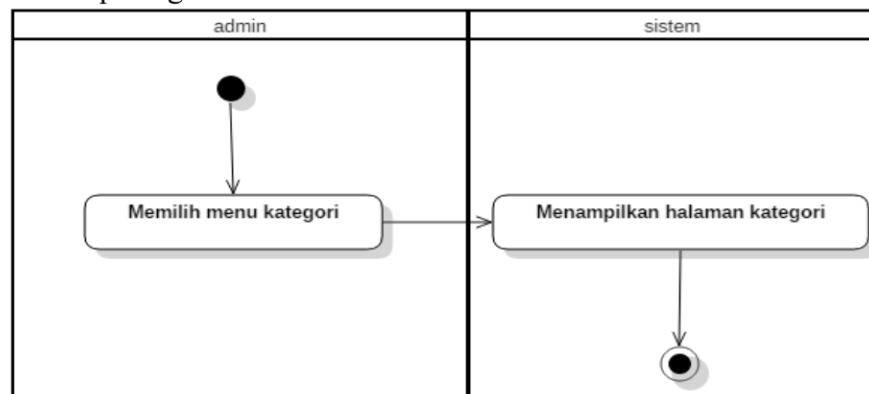
Diagram ini menjelaskan jalanya alur sistem ketika *admin* mengakses halaman berita. Pada halaman ini terdapat informasi kesehatan yang berkaitan dengan penyakit hepatitis serta informasi umum lainnya yang dapat dibaca oleh *user*. *Admin* dapat melakukan tambah, edit, serta hapus pada data berita. Adapun *activity* diagram berita (*admin*) dapat dilihat pada gambar 4.17



Gambar 4.17 Activity diagram berita (Admin)

17. Activity diagram kategori berita (Admin)

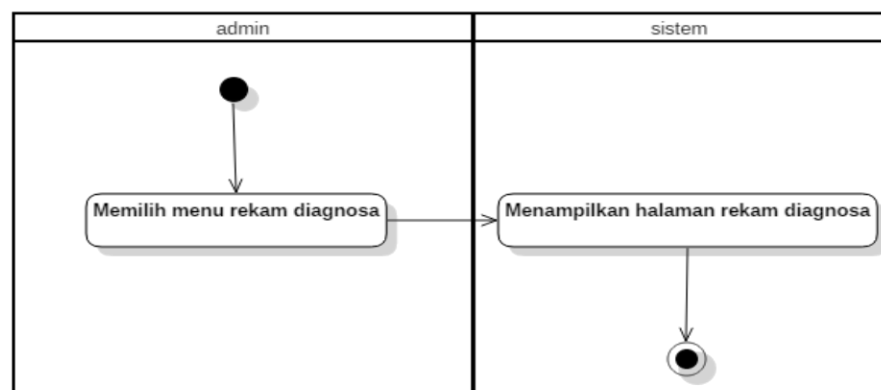
Diagram ini menjelaskan alur sistem ketika *admin* mengakses halaman kategori berita. Terdapat kategori kesehatan dan umum yang berkaitan dengan penyakit hepatitis serta mempermudah *user* dalam mencari berita yang akan dibaca. *Admin* dapat mengelola data kategori berita seperti tambah, edit, dan hapus. Adapun *activity* diagram kategori berita (*admin*) dapat dilihat pada gambar 4.18



Gambar 4.18 Activity diagram kategori berita (Admin)

18. Activity diagram rekam diagnosis (Admin)

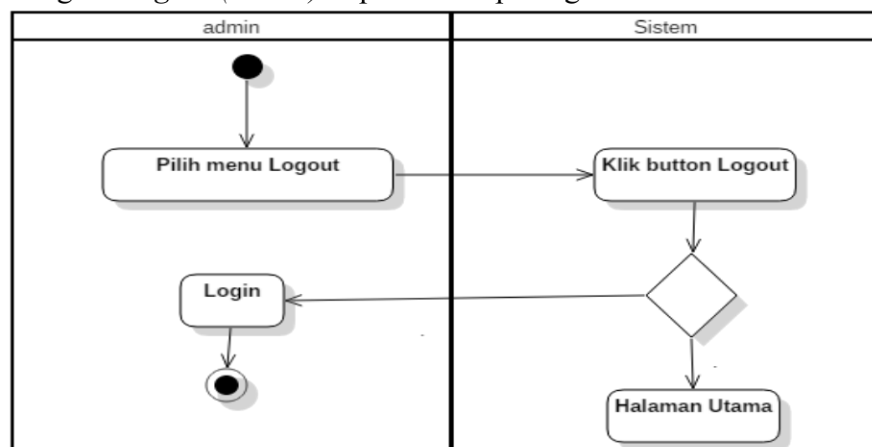
Diagram ini menjelaskan jalanya alur sistem ketika *admin* mengakses halaman rekam diagnosis. Data ini berisi *history* diagnosis penyakit yang telah dilakukan *user*, data ini disimpan untuk kepentingan *user* karna selanjutnya dapat dilihat detail mengenai penyakit yang diderita serta dapat dicetak. *Admin* tidak dapat mengubah data ini. Adapun *activity* diagram rekam diagnosis (*admin*) dapat dilihat pada gambar 4.19



Gambar 4.19 Activity diagram rekam diagnosis (Admin)

19. Activity diagram logout (Admin)

Diagram ini menjelaskan alur sistem ketika *admin* keluar dari sistem, *session admin* akan berakhir dan diarahkan ke halaman *login* awal. *Admin* dapat memasukkan *username* dan *password* untuk masuk lagi kedalam sistem dan mengelola sistem. Adapun *activity* diagram *logout* (*admin*) dapat dilihat pada gambar 4.20



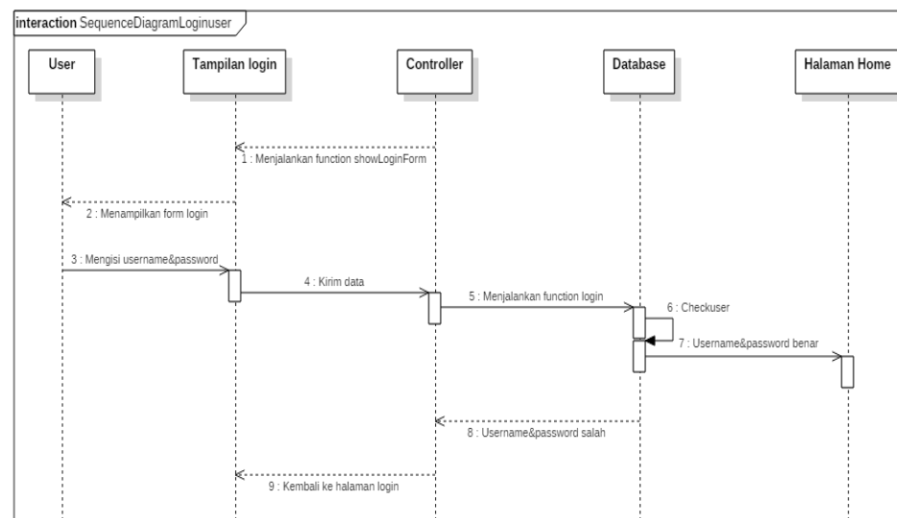
Gambar 4.20 Activity diagram logout (Admin)

4.2.3 Sequence Diagram

Sequence diagram mendeskripsikan, mengembangkan komunikasi, menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, tampilan dan sebagainya) secara beruntun. Berikut *sequence diagram* aplikasi sistem pakar diagnosis awal penyakit hepatitis :

1. *Sequence diagram login (User)*

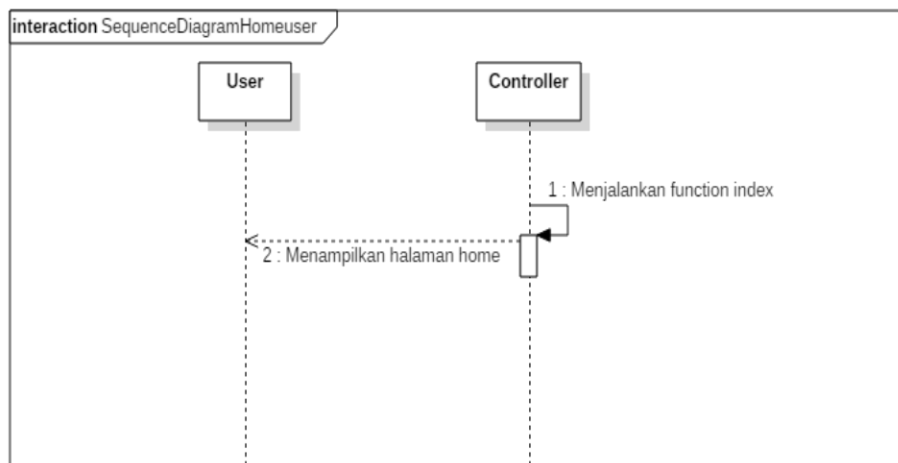
Diagram *login* menjelaskan tentang jalanya aplikasi untuk proses *login*, *user* memasukan *username* dan *password*. Ketika *username* dan *password* sesuai maka secara otomatis masuk ke tampilan beranda dan jika salah maka akan kembali ke halaman *login*. Adapun *sequence diagram login (user)* dapat dilihat pada gambar 4.21



Gambar 4.21 *Sequence diagram login (User)*

2. *Sequence diagram beranda (User)*

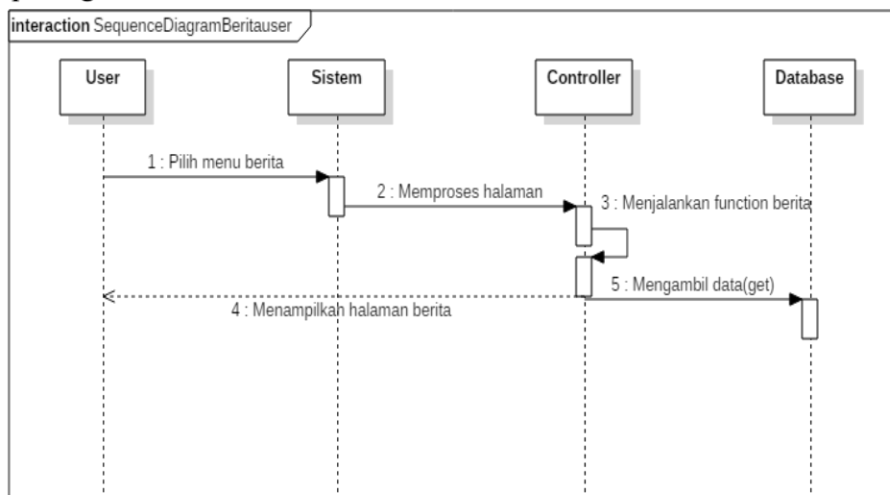
Diagram beranda menjelaskan jalanya alur sistem ketika *user* mengakses halaman beranda. Halaman ini akan tampil diawal saat *user* mengakses saat mengakses sistem, berisi tombol mulai diagnosis, berita seputar kesehatan serta menu lainnya. Adapun *sequence diagram beranda (user)* dapat dilihat pada gambar 4.22



Gambar 4.22 *Sequence diagram beranda (User)*

3. *Sequence diagram berita (User)*

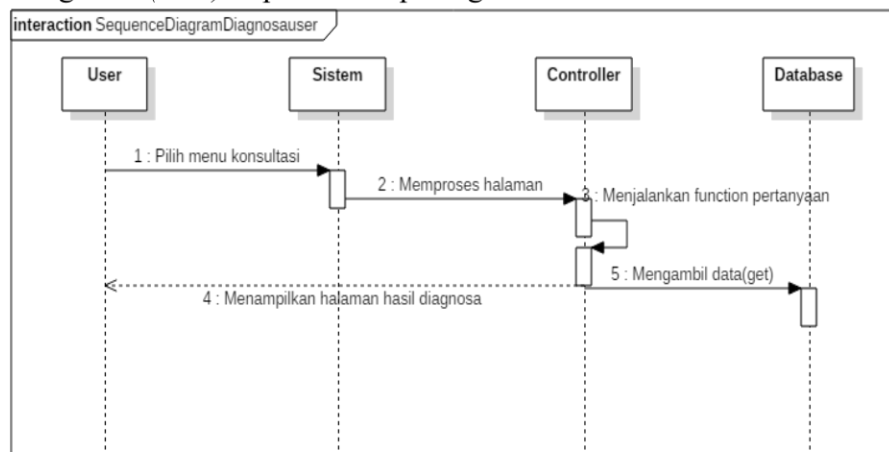
Diagram berita menjelaskan alur sistem untuk halaman berita, didalam halaman ini terdapat berita terkait penyakit hepatitis dan kesehatan. Adapun *sequence diagram berita (user)* dapat dilihat pada gambar 4.23



Gambar 4.23 *Sequence diagram berita (User)*

4. *Sequence diagram diagnosis (User)*

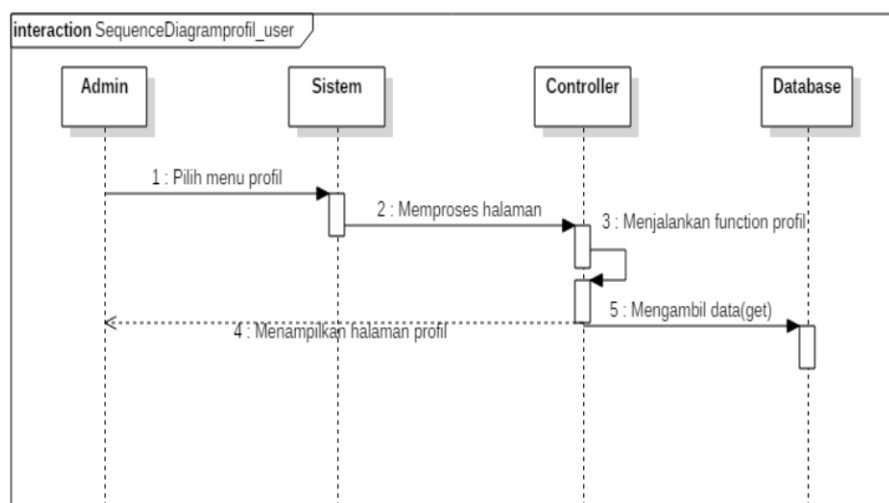
Diagram diagnosis menjelaskan alur sistem ketika *user* mengakses halaman diagnosis, dalam halaman ini *user* diminta untuk mengisi gejala yang dirasa sehingga akan dihasilkan diagnosis jenis penyakit yang diderita. Adapun *sequence diagram diagnosis (user)* dapat dilihat pada gambar 4.24



Gambar 4.24 *Sequence diagram diagnosis (User)*

5. *Sequence diagram profil (User)*

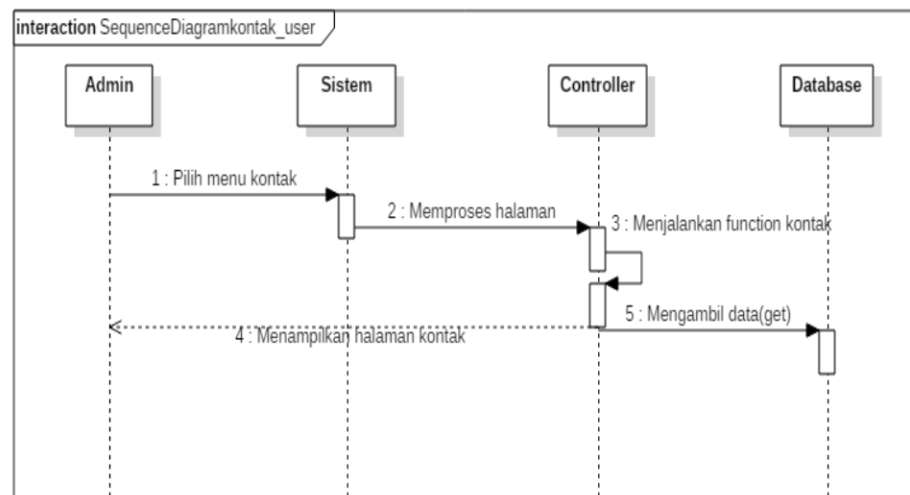
Diagram profil menjelaskan jalanya alur sistem ketika *user* mengakses halaman profil, halaman profil berisi detail penggunaan sistem pakar serta fungsi dibuatnya sistem pakar. Adapun *sequence diagram profil (user)* dapat dilihat pada gambar 4.25



Gambar 4.25 *Sequence diagram profil (User)*

6. Sequence diagram kontak (User)

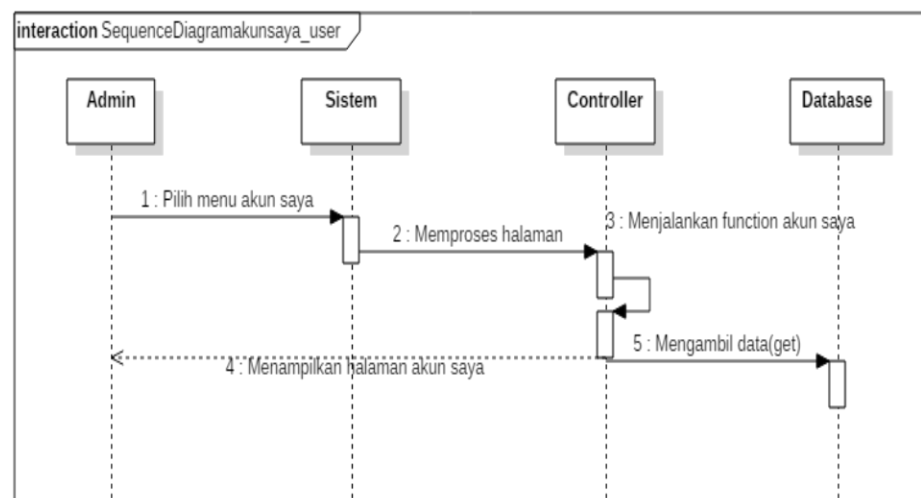
Diagram kontak menjelaskan jalanya alur sistem ketika *user* mengakses halaman kontak, di mana terdapat kontak yang dapat dihubungi oleh *user* jika ada pertanyaan atau bantuan saat menggunakan aplikasi. Adapun *sequence* diagram kontak (*user*) dapat dilihat pada gambar 4.26



Gambar 4.26 Sequence diagram kontak (User)

7. Sequence diagram akun saya (User)

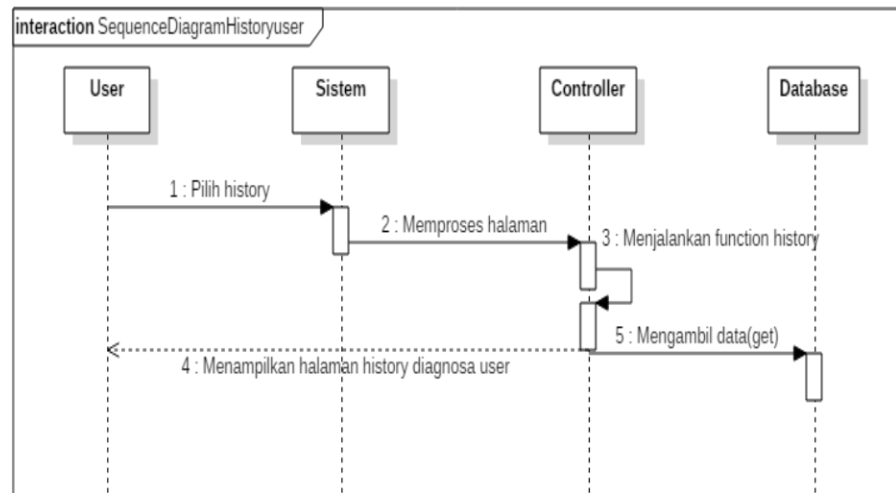
Diagram akun saya menjelaskan jalanya alur sistem ketika *user* mengakses halaman akun saya, pada halaman ini *user* dapat merubah informasi pribadi seperti *password*, dan nomor hp. Adapun *sequence* diagram akun saya (*user*) dapat dilihat pada gambar 4.27



Gambar 4.27 Sequence diagram akun saya (User)

8. *Sequence diagram* hasil diagnosis (*User*)

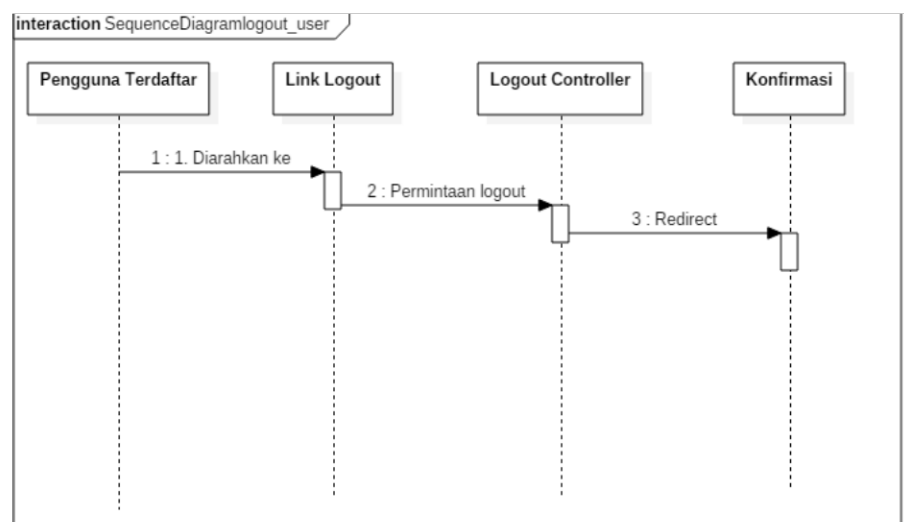
Diagram hasil diagnosis menjelaskan jalanya alur sistem pada aksi lihat diagnosis, *user* yang telah melakukan proses diagnosis dapat melihat hasil diagnosisnya. Adapun *sequence diagram* hasil diagnosis (*user*) dapat dilihat pada gambar 4.28



Gambar 4.28 *Sequence diagram* hasil diagnosis (*User*)

9. *Sequence diagram* logout (*User*)

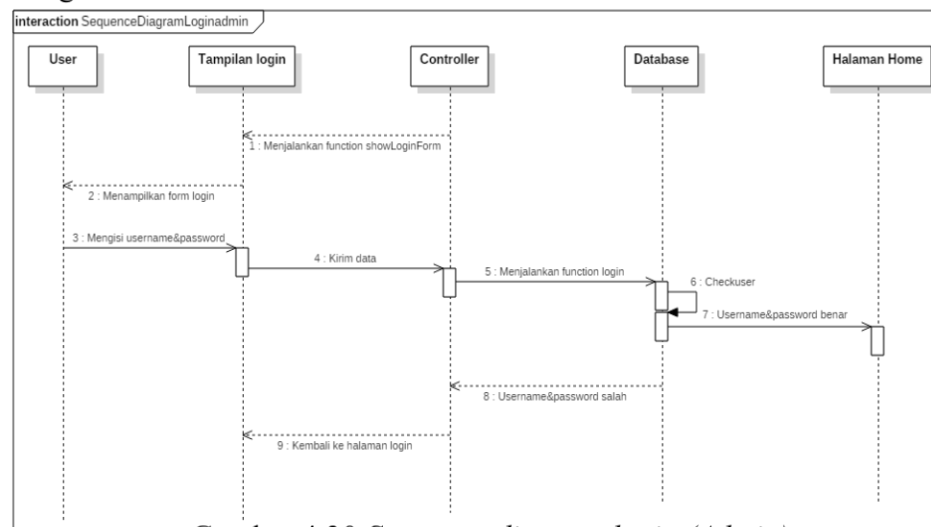
Diagram *logout* menjelaskan jalanya alur sistem ketika *user* mengakses fitur *logout*, *user* akan kembali kehalaman awal dan keluar dari akun nya. Adapun *sequence diagram* *logout* (*user*) dapat dilihat pada gambar 4.29



Gambar 4.29 *Sequence diagram* logout (*User*)

10. Sequence diagram login (Admin)

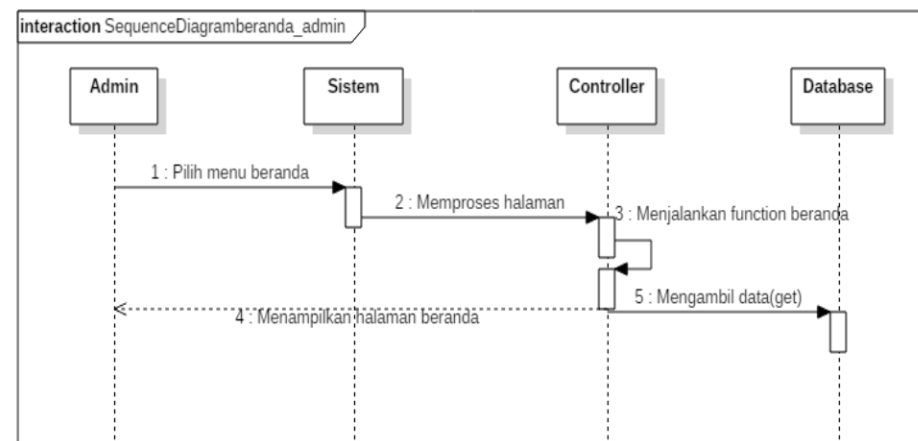
Diagram ini menjelaskan tentang jalanya aplikasi untuk proses *login, admin* memasukkan *username* dan *password*. Ketika *username* dan *password* sesuai maka secara otomatis masuk ke tampilan beranda dan jika salah maka akan kembali ke halaman *login*. Adapun *sequence diagram login (admin)* dapat dilihat pada gambar 4.30



Gambar 4.30 Sequence diagram login (Admin)

11. Sequence diagram beranda (Admin)

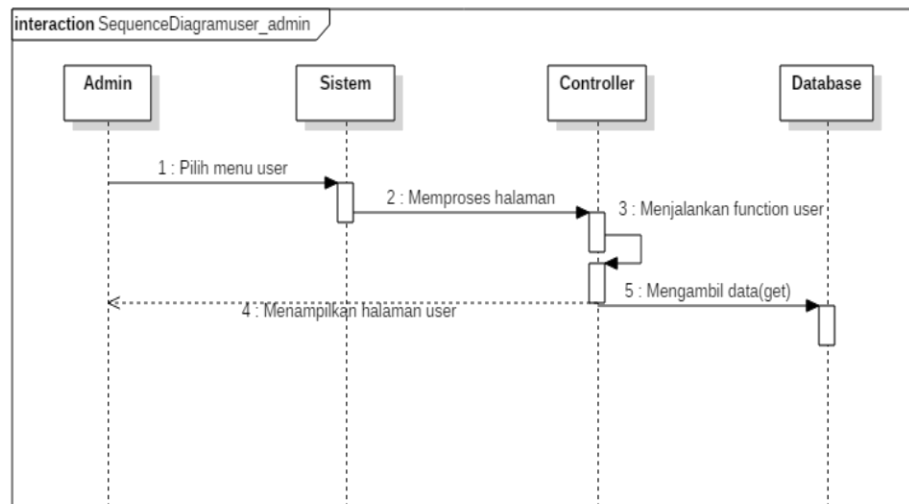
Diagram ini menjelaskan tentang jalanya aplikasi untuk tampilan beranda (*dashboard*), pada beranda terdapat informasi singkat tentang sistem seperti *history* diagnosis terbaru, dll. Adapun *sequence diagram beranda (admin)* dapat dilihat pada gambar 4.31



Gambar 4.31 Sequence diagram login (Admin)

12. Sequence diagram user (Admin)

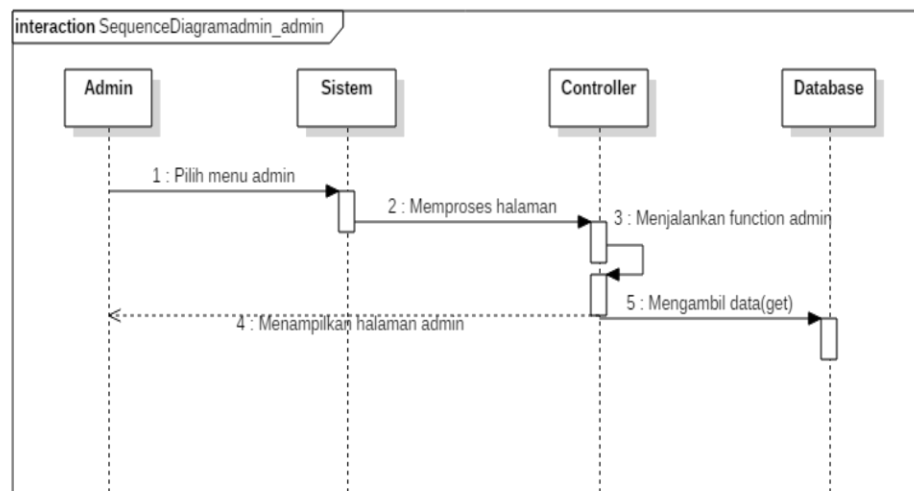
Diagram ini menjelaskan jalanya alur sistem ketika menampilkan halaman *user*. Di mana *admin* dapat mengolah sumber daya data *user* mulai dari tambah, edit, dan hapus. Adapun *sequence diagram user (admin)* dapat dilihat pada gambar 4.32



Gambar 4.32 Sequence diagram user (Admin)

13. Sequence diagram admin (Admin)

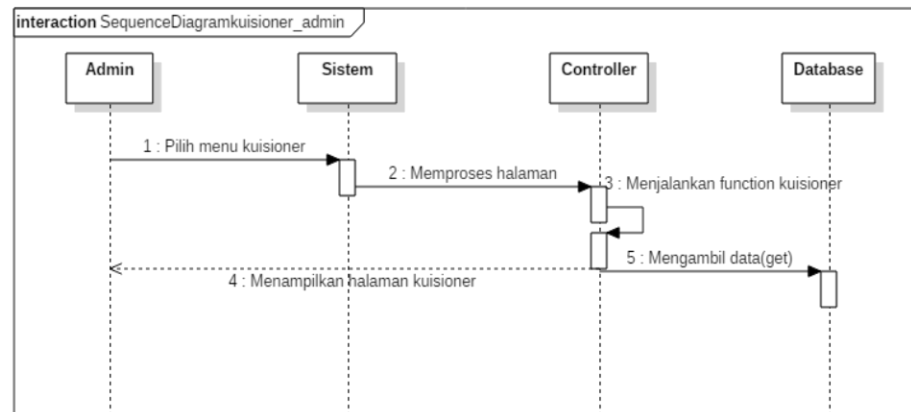
Diagram ini menjelaskan alur sistem ketika *admin* mengakses halaman *admin*, di mana *admin* dapat mengelola sumber daya sistem berupa data *admin* yang dapat di tambah, edit, dan hapus. Adapun *sequence diagram admin (admin)* dapat dilihat pada gambar 4.33



Gambar 4.33 Sequence diagram admin (Admin)

14. Sequence diagram gejala (Admin)

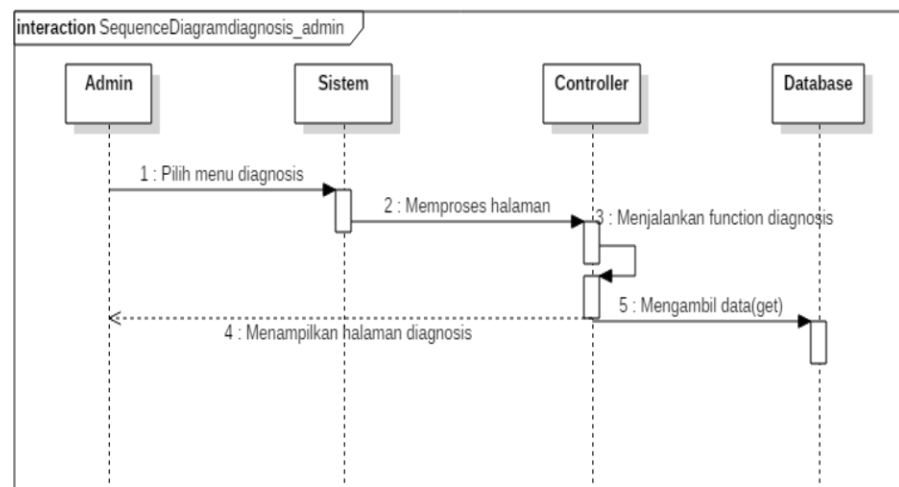
Diagram ini menjelaskan jalannya alur sistem ketika *admin* mengakses halaman gejala. Pada halaman ini, terdapat daftar pertanyaan yang bersangkutan dengan gejala penyakit hepatitis. *Admin* dapat melakukan tambah, edit, serta hapus pada data gejala. Adapun *sequence* diagram gejala (*admin*) dapat dilihat pada gambar 4.34



Gambar 4.34 Sequence diagram gejala (Admin)

15. Sequence diagram diagnosis (Admin)

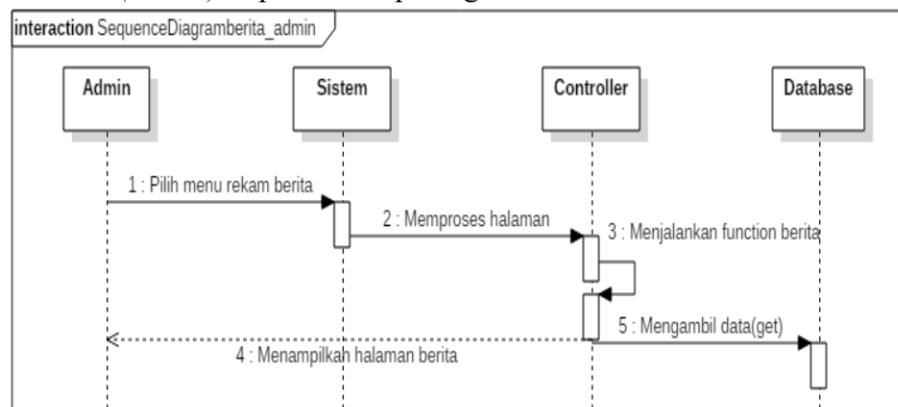
Diagram ini menjelaskan alur sistem ketika *admin* mengakses halaman diagnosis. *Admin* dapat mengelola sumber daya data diagnosis yang berisi jenis penyakit hepatitis serta dapat melakukan olah data seperti tambah, edit, dan hapus. Adapun *sequence* diagram diagnosis (*admin*) dapat dilihat pada gambar 4.35



Gambar 4.35 Sequence diagram diagnosis (Admin)

16. Sequence diagram berita (Admin)

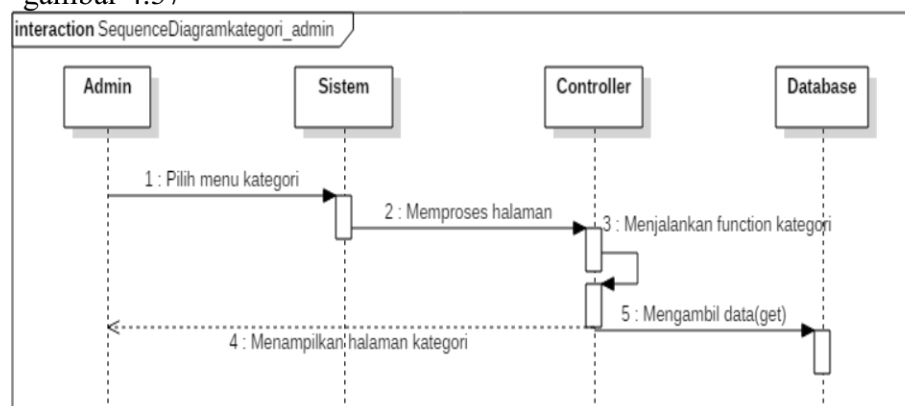
Diagram ini menjelaskan jalanya alur sistem ketika *admin* mengakses halaman berita. Pada halaman ini terdapat informasi kesehatan yang berkaitan dengan penyakit hepatitis serta informasi umum lainnya yang dapat dibaca oleh *user*. *Admin* dapat melakukan tambah, edit, serta hapus pada data berita. Adapun *sequence* diagram berita (*admin*) dapat dilihat pada gambar 4.36



Gambar 4.36 Sequence diagram berita (Admin)

17. Sequence diagram kategori berita (Admin)

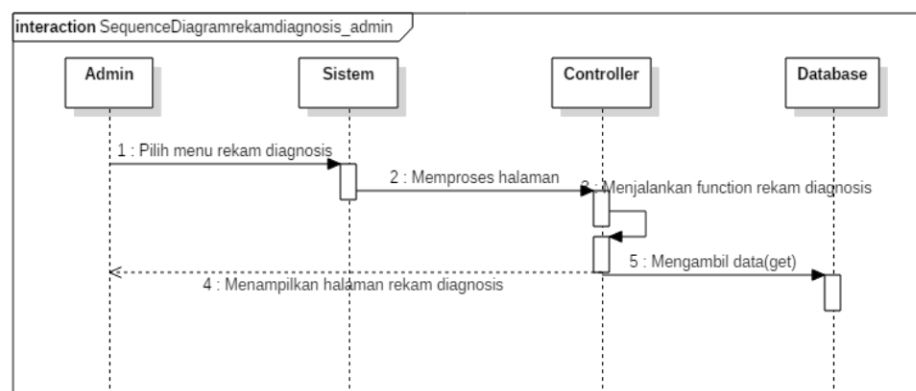
Diagram ini menjelaskan alur sistem ketika *admin* mengakses halaman kategori berita. Terdapat kategori kesehatan dan umum yang berkaitan dengan penyakit hepatitis serta mempermudah *user* dalam mencari berita yang akan dibaca. *Admin* dapat mengelola data kategori berita seperti tambah, edit, dan hapus. Adapun *sequence* diagram kategori berita (*admin*) dapat dilihat pada gambar 4.37



Gambar 4.37 Sequence diagram kategori berita (Admin)

18. Sequence diagram rekam diagnosis (Admin)

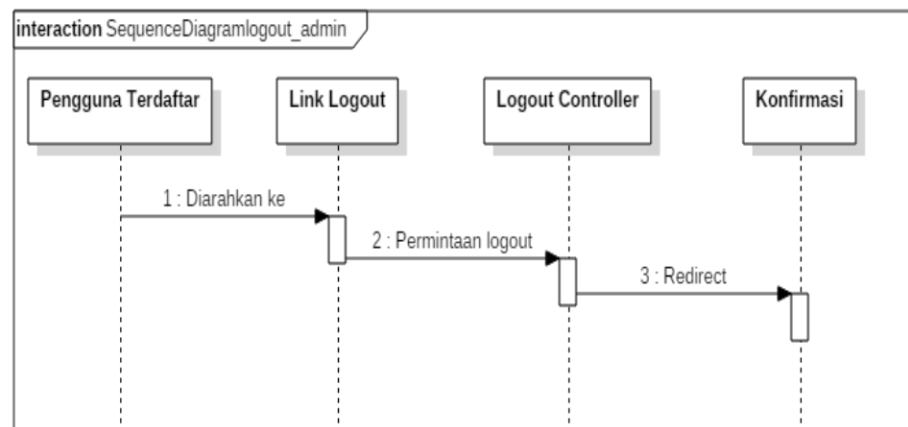
Diagram ini menjelaskan jalanya alur sistem ketika *admin* mengakses halaman rekam diagnosis. Data ini berisi *history* diagnosis penyakit yang telah dilakukan *user*, data ini disimpan untuk kepentingan *user* karna selanjutnya dapat dilihat detail mengenai penyakit yang diderita serta dapat dicetak. *Admin* tidak dapat mengubah data ini. Adapun *sequence* diagram rekam diagnosis (*admin*) dapat dilihat pada gambar 4.38



Gambar 4.38 Sequence diagram rekam diagnosis (Admin)

19. Sequence diagram logout (Admin)

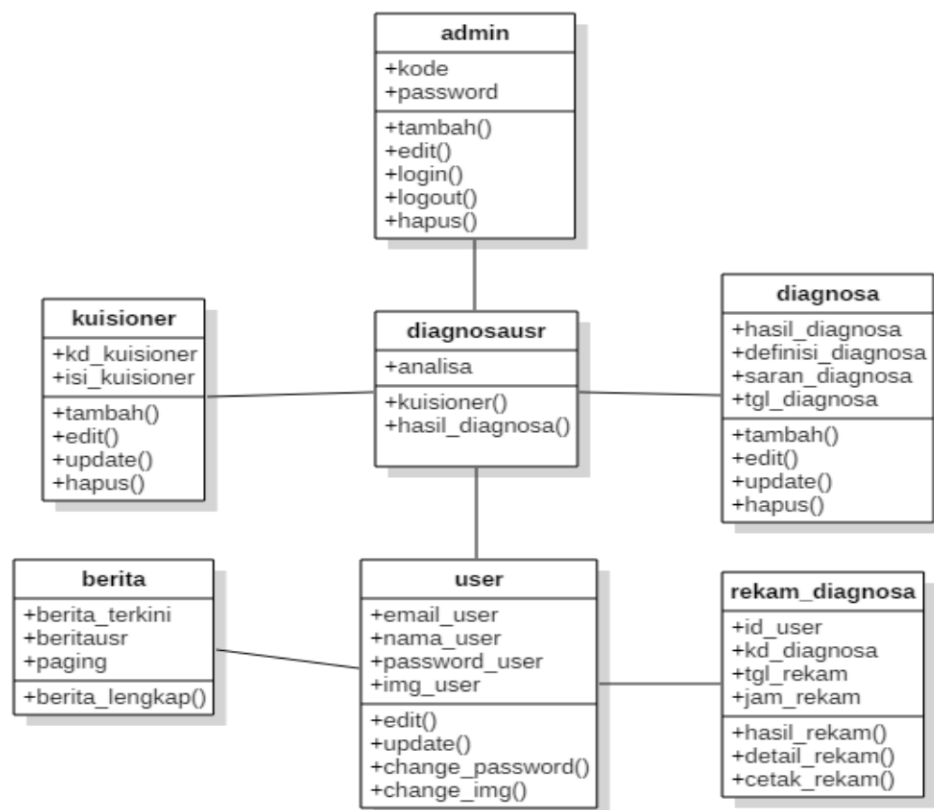
Diagram ini menjelaskan alur sistem ketika *admin* keluar dari sistem, *session admin* akan berakhir dan diarahkan ke halaman *login* awal. *Admin* dapat memasukkan *username* dan *password* untuk masuk lagi kedalam sistem dan mengelola sumber daya. Adapun *sequence* diagram *logout* (*admin*) dapat dilihat pada gambar 4.39



Gambar 4.39 Sequence diagram logout(Admin)

4.2.4 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur ini meliputi atribut, *method*, atau fungsi yang ada pada masing-masing *class*. *Class diagram* dibuat agar pembuat program membuat kelas-kelas sesuai dengan rancangan yang ada, sehingga antara dokumentasi perancangan dan perangkat lunak memiliki kesesuaian. Pada gambar 4.40 menjelaskan tentang *class diagram* yang terdapat pada aplikasi sistem pakar penyakit hepatitis.



Gambar 4.40 *Class diagram*

Class diagram diatas diperoleh dari:

- Pendefinisian *use case (controller)*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada sistem.
- Pendefinisian data (*model*), kelas yang digunakan untuk membungkus data menjadi kesatuan yang diambil atau disimpan ke *database*.

4.3 Perancangan *Database*.

Dalam perancangan *database* aplikasi sistem pakar terdapat beberapa struktur, tabel basis data yang akan digunakan sebagai penyimpanan data dan informasi. Berikut tabel yang digunakan:

Nama *database*: dbbelajarci02

4.3.1 Perancangan tabel

Dalam perancangan aplikasi sistem pakar diagnosis awal hepatitis menspesifikasikan tabel-tabel yang terdapat pada *database*. Berikut spesifikasi tabel-tabel yang dibuat.

Tabel 4.4 Struktur Tabel *admin*

Nama field	Tipe	Ukuran	Keterangan
kd_admin	varchar	10	<i>Primary Key</i>
nama_admin	varchar	50	
pswd_admin	varchar	50	
hp_admin	varchar	13	
email_admin	varchar	30	
img_admin	varchar	100	

Tabel 4.5 Struktur Tabel Berita

Nama field	Tipe	Ukuran	Keterangan
id_berita	int	11	<i>Primary Key</i>
kd_admin	varchar	10	<i>Foreign Key</i>
id_kategori	int		<i>Foreign Key</i>
st_berita	varchar	64	
tgl_berita	date		
jam_berita	time		
judul_berita	varchar	100	
isi_berita	text		
img_berita	varchar	100	

Tabel 4.6 Tabel diagnosis

Nama field	Tipe	Ukuran	Keterangan
kd_diagnosis	varchar	5	<i>Primary Key</i>
hasil_diagnosis	text		
defini_diagnosis	text		
saran_diagnosis	text		
tgl_diagnosis	date		
sumber_diagnosis	varchar	100	

Tabel 4.7 Struktur Tabel kategori

Nama field	Tipe	Ukuran	Keterangan
id_kategori	int	4	<i>Primary Key</i>
nama_kategori	varchar	30	

Tabel 4.8 Tabel kuisisioner

Nama field	Tipe	Ukuran	Keterangan
kd_kuisisioner	Char	5	<i>Primary Key</i>
isi_kuisisioner	Text		

Tabel 4.9 Tabel rekam diagnosis

Nama field	Tipe	Ukuran	Keterangan
id_rekam	int	11	<i>Primary Key</i>
id_user	int	11	<i>Foreign Key</i>
kd_diagnosis	varchar	5	<i>Foreign Key</i>
tgl_rekam	date		
jam_rekam	time		

Tabel 4.10 Struktur Tabel user

Nama field	Tipe	Ukuran	Keterangan
id_user	int	12	<i>Primary Key</i>
nama_user	varchar	50	
email_user	varchar	30	
pswd_user	varchar	50	
hp_user	varchar	13	

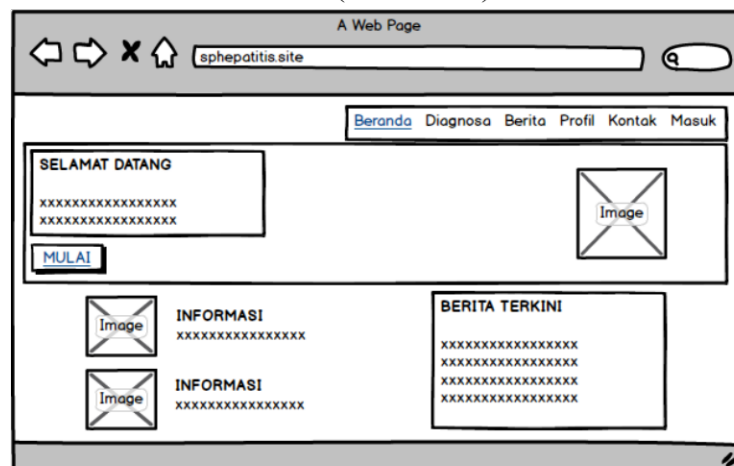
img_user	varchar	100	
----------	---------	-----	--

4.4 Perancangan Antar Muka.

Pada tahap ini dilakukan perancangan *user interface* untuk mempermudah pengguna untuk berinteraksi dengan sistem. Perancangan *user interface* meliputi perancangan *interface input* dan *interface output*. *Interface input* digunakan untuk membantu pengguna dalam memasukan data kedalam *database*, sedangkan *interface output* digunakan untuk menampilkan data-data yang tersimpan didalam *database* untuk memperoleh informasi yang berguna bagi pemakainya. Berikut adalah *user interface* pada sistem pakar penyakit hepatitis.

1. Halaman beranda (*user*)

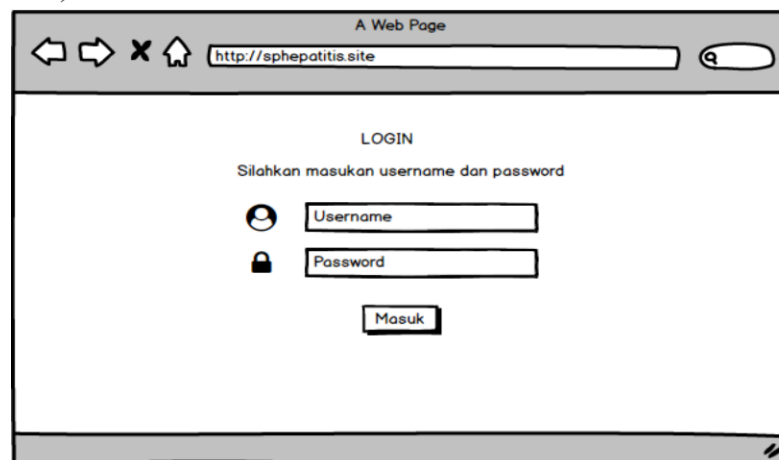
Gambar 4.41 merupakan desain halaman beranda *user*, dimana halaman ini adalah halaman yang pertama muncul ketika *user* mengakses sistem pakar dan selanjutnya diarahkan ke proses *login*. Desain halaman beranda *user* dibuat berdasarkan *activity diagram* beranda *user* pada Gambar 4.3 dan *sequence diagram* beranda *user* pada Gambar 4.22 serta menggunakan data dari Tabel user (Tabel 4.10).



Gambar 4.41 Rancangan halaman beranda *user*

2. Halaman *login (user)*

Tampilan perancangan dari halaman *login user* ditunjukkan pada gambar 4.42, pada halaman ini *user* memasukan *username* dan *password* sesuai dengan akun yang telah dibuat sebelumnya. Desain halaman ini dibuat berdasarkan *activity diagram login user* pada Gambar 4.2 dan *sequence diagram login user* pada Gambar 4.21 serta data dari Tabel user (Tabel 4.10).



A Web Page
http://sphepatitis.site

LOGIN

Silahkan masukan username dan password

Username

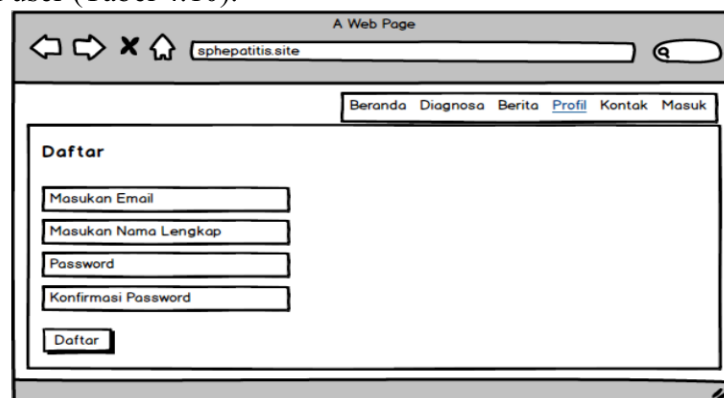
Password

Masuk

Gambar 4.42 Rancangan halaman *login user*

3. Halaman daftar (*user*)

Tampilan perancangan dari halaman daftar *user* ditunjukkan pada gambar 4.43, halaman ini berisi *field* yang harus diisi oleh *user* untuk memperoleh akun yang selanjutnya digunakan untuk proses diagnosis. Desain halaman ini dibuat berdasarkan *activity diagram* daftar *user* pada Gambar 4.8 dan *sequence diagram* daftar *user* pada Gambar 4.27 serta data dari Tabel user (Tabel 4.10).



A Web Page
sphepatitis.site

Beranda Diagnosa Berita Profil Kontak Masuk

Daftar

Masukan Email

Masukan Nama Lengkap

Password

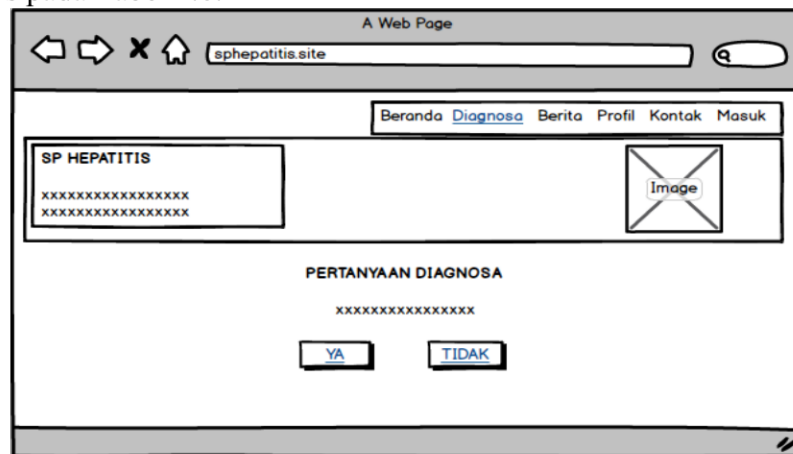
Konfirmasi Password

Daftar

Gambar 4.43 Rancangan halaman daftar *user*

4. Halaman diagnosis (*user*)

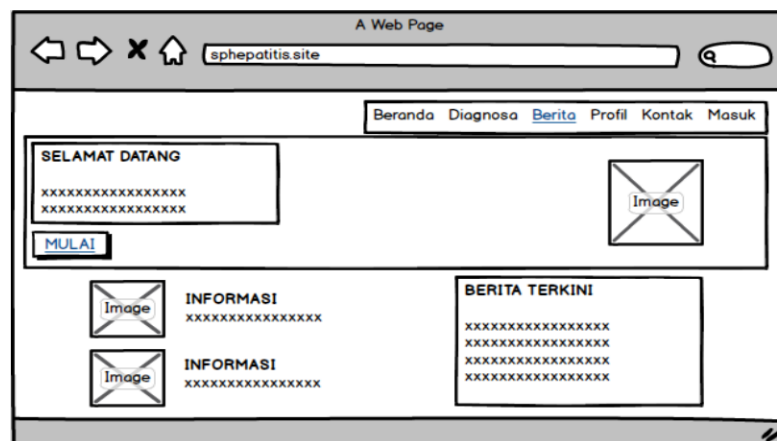
Tampilan perancangan dari halaman diagnosis *user* ditunjukkan pada gambar 4.44, halaman ini berupa proses diagnosis yang berisi daftar pertanyaan sesuai dengan gejala yang dirasakan *user*. Desain halaman ini dibuat berdasarkan *activity diagram* diagnosis *user* pada Gambar 4.4 dan *sequence diagram* diagnosis pada Gambar 4.24 serta data dari Tabel diagnosis pada Tabel 4.6.



Gambar 4.44 Rancangan halaman diagnosis *user*

5. Halaman berita (*user*)

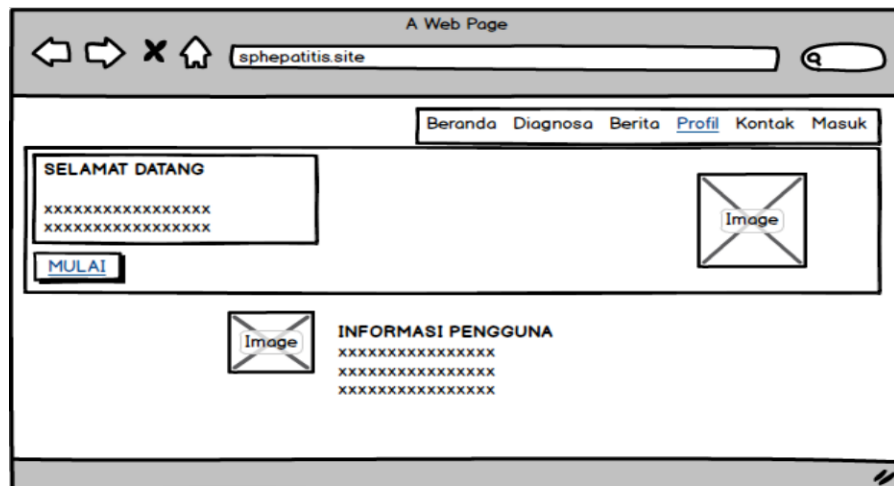
Gambar 4.45 merupakan desain halaman berita *user*, dimana terdapat informasi kesehatan dan umum yang dapat dibaca *user*. Halaman ini dibuat berdasarkan *activity diagram* berita *user* pada Gambar 4.5, *sequence diagram* berita *user* pada Gambar 4.23, serta data dari Tabel berita pada Tabel 4.5.



Gambar 4.45 Rancangan halaman berita *user*

6. Halaman profil *user*

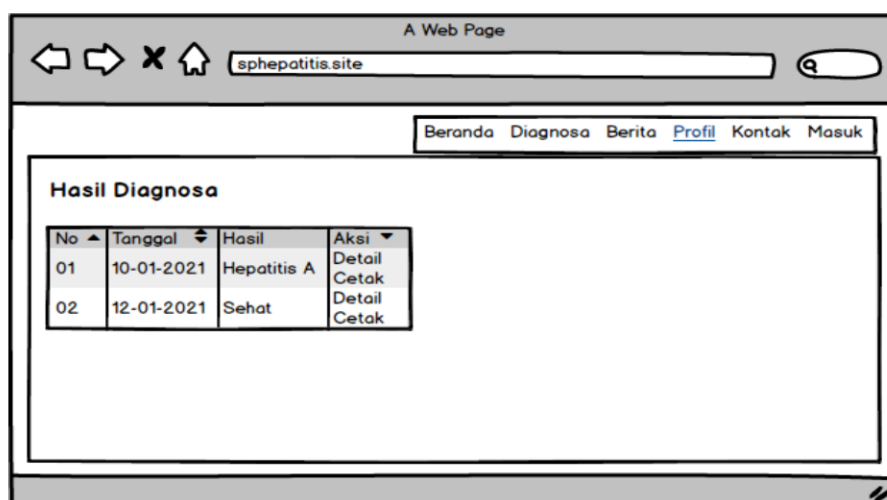
Tampilan perancangan dari halaman profil *user* ditunjukkan pada gambar 4.46, halaman ini berisi profil mengenai sistem pakar serta cara penggunaannya. Halaman ini dibuat berdasarkan *activity diagram* profil *user* pada Gambar 4.6, dan *sequence diagram* profil *user* Gambar 4.25.



Gambar 4.46 Rancangan halaman profil *user*

7. Halaman hasil diagnosis *user*

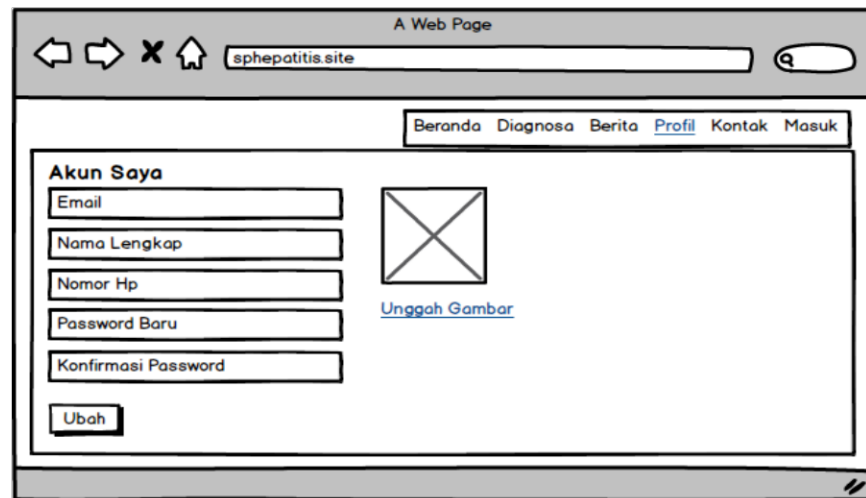
Gambar 4.47 merupakan desain halaman hasil diagnosis *user*, pada halaman ini terdapat *history* diagnosis yang telah dilakukan *user* sebelumnya. Halaman ini dibuat berdasarkan *activity diagram* hasil diagnosis *user* pada Gambar 4.9, *sequence diagram* hasil diagnosis *user* pada Gambar 4.28, dan data dari Tabel rekam diagnosis pada Tabel 4.9.



Gambar 4.47 Rancangan halaman diagnosis *user*

8. Halaman akun saya (*user*)

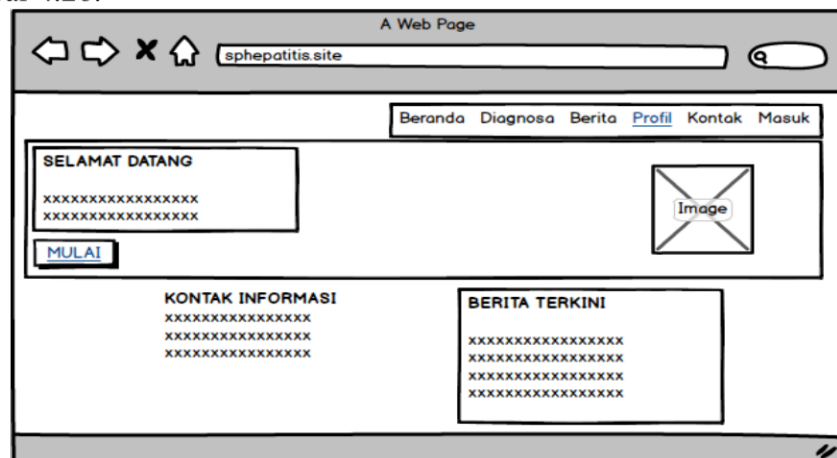
Tampilan perancangan dari halaman daftar *user* ditunjukkan pada gambar 4.48, pada halaman ini *user* dapat merubah informasi pribadi seperti nomor hp, *email*, dll. Halaman ini dibuat berdasarkan *activity diagram* akun saya pada Gambar 4.8, *sequence diagram* akun saya pada Gambar 4.27, serta data pada Tabel *user* pada Tabel 4.10.



Gambar 4.48 Rancangan halaman akun saya (*user*)

9. Halaman kontak *user*

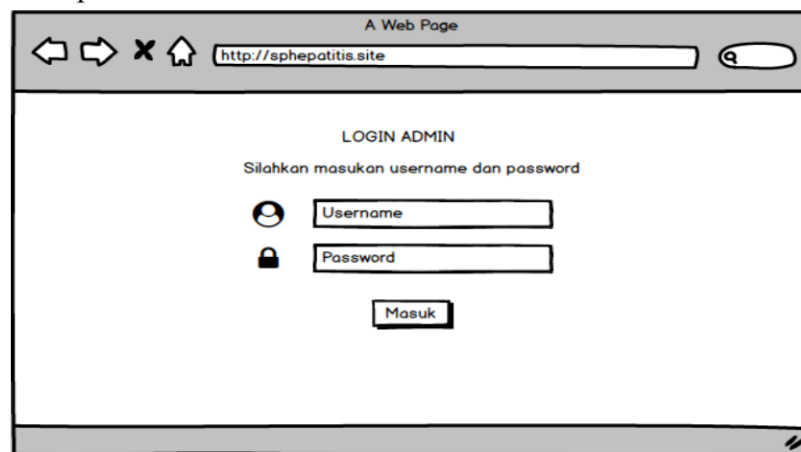
Tampilan perancangan dari halaman kontak *user* ditunjukkan pada gambar 4.49, halaman ini terdapat kontak informasi yang dapat dihubungi oleh *user* jika ada pertanyaan. Halaman ini dibuat berdasarkan *activity diagram* kontak *user* pada Gambar 4.7, *sequence diagram* kontak *user* pada Gambar 4.26.



Gambar 4.49 Rancangan halaman kontak *user*

10. Halaman *login admin*

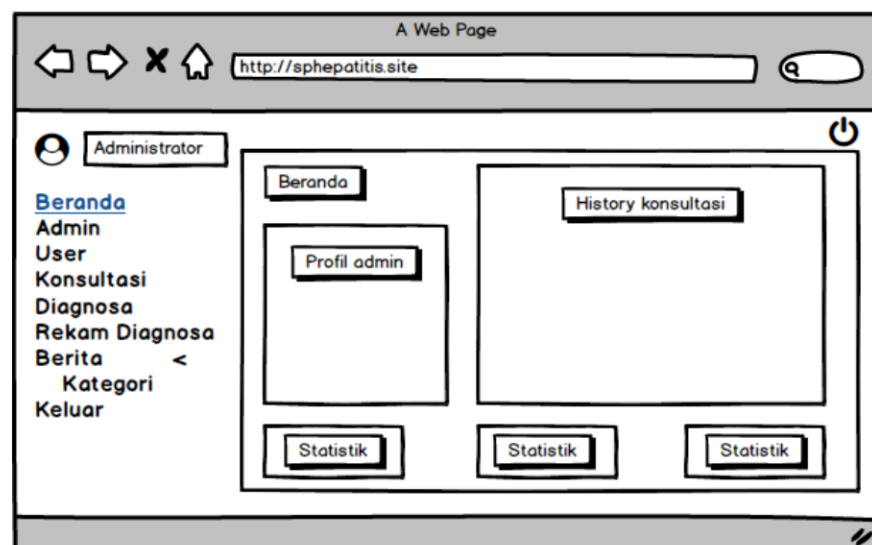
Gambar 4.50 merupakan desain halaman *login admin*, pada halaman ini *admin* diharuskan mengisi *username* dan *password* untuk masuk kedalam sistem. Halaman ini dibuat berdasarkan *activity diagram login admin* pada Gambar 4.12, *sequence diagram* pada Gambar 4.30, serta data Tabel admin pada Tabel 4.4.



Gambar 4.50 Rancangan halaman *login admin*

11. Halaman beranda *admin*

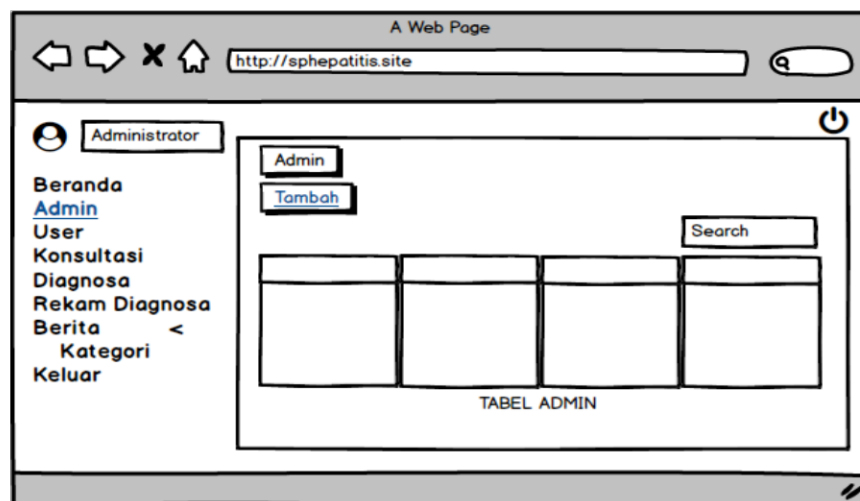
Gambar 4.51 merupakan desain halaman beranda *admin*, halaman ini berisi informasi singkat mengenai sistem yang disajikan dalam bentuk grafik. Halaman ini dibuat berdasarkan *activity diagram* pada Gambar 4.11, dan *sequence diagram* pada Gambar 4.31



Gambar 4.51 Rancangan halaman beranda *admin*

12. Halaman *admin (admin)*

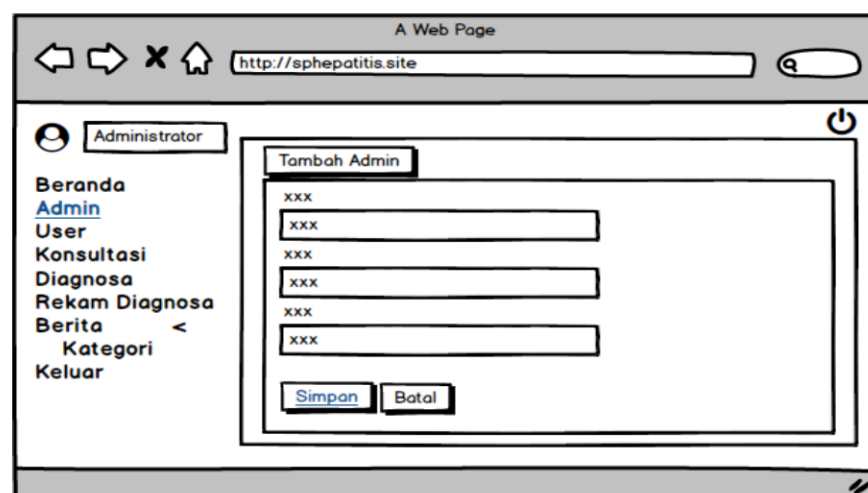
Tampilan perancangan dari halaman *admin* ditunjukkan pada gambar 4.52, halaman ini berisi *data table admin* yang dapat ditambah, edit dan hapus. Halaman ini dibuat berdasarkan *activity diagram admin* pada Gambar 4.14, *sequence diagram admin* pada Gambar 4.33, serta data Tabel Admin pada Tabel 4.4.



Gambar 4.52 Rancangan halaman *admin (admin)*

13. Halaman tambah *admin*

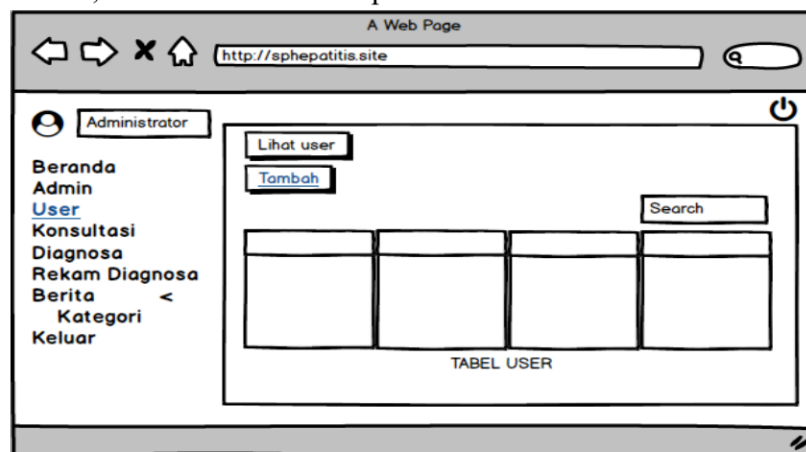
Tampilan perancangan dari halaman tambah *admin* ditunjukkan pada gambar 4.5, pada halaman ini terdapat *field* yang harus diisi jika akan menambahkan akun *admin* baru. Halaman ini dibuat berdasarkan data Tabel Admin pada Tabel 4.4.



Gambar 4.53 Rancangan halaman tambah *admin*

14. Halaman *user (admin)*

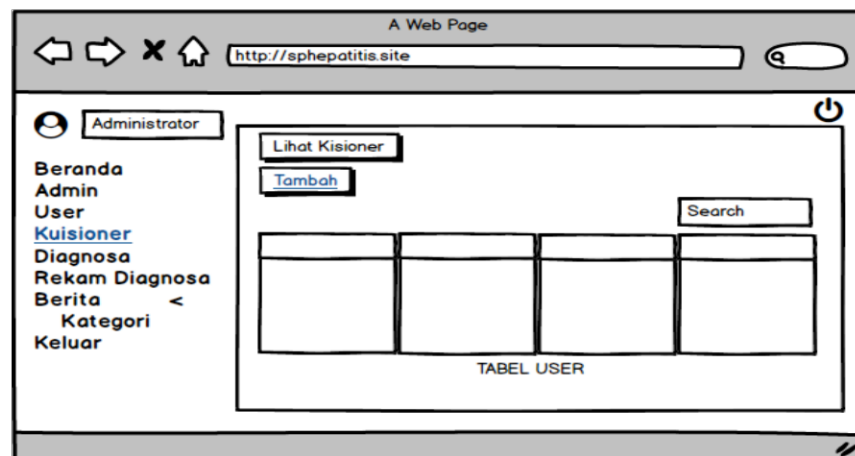
Gambar 4.54 merupakan desain halaman *user admin*, halaman ini berisi *data table* yang memuat informasi akun *user*. *Admin* hanya dapat melakukan aksi detail dan hapus. Halaman ini dibuat berdasarkan *activity diagram user admin* pada Gambar 4.13, *sequence diagram user admin* pada Gambar 4.32, dan data Tabel User pada Tabel 4.10.



Gambar 4.54 Rancangan halaman *user (admin)*

15. Halaman *kuisisioner (admin)*

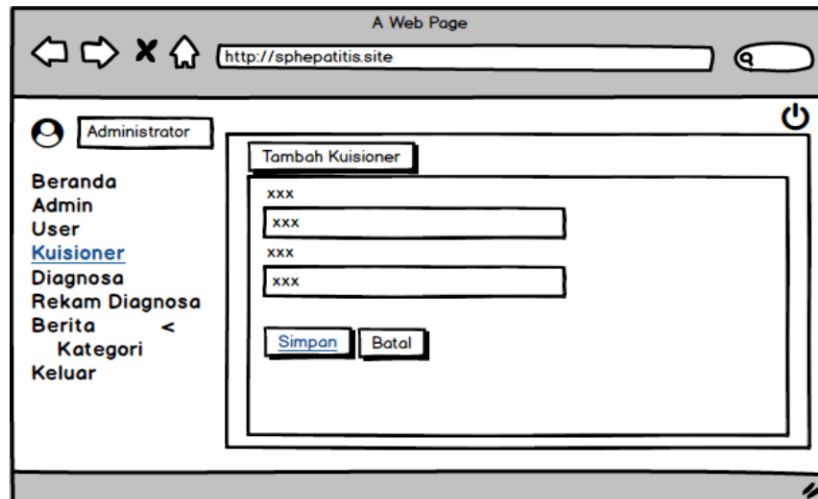
Gambar 4.55 merupakan desain halaman *kuisisioner admin*, halaman ini terdapat *data table* *kuisisioner* yang berisi pertanyaan gejala yang akan diajukan pada *user*. *Admin* dapat melakukan aksi tambah, edit, dan hapus. Halaman ini dibuat berdasarkan *activity diagram kuisisioner* pada Gambar 4.15, *sequence diagram kuisisioner* pada Gambar 4.34, dan data Tabel *Kuisisioner* pada Tabel 4.8.



Gambar 4.55 Rancangan halaman *kuisisioner (admin)*

16. Halaman tambah kuisioner (*admin*)

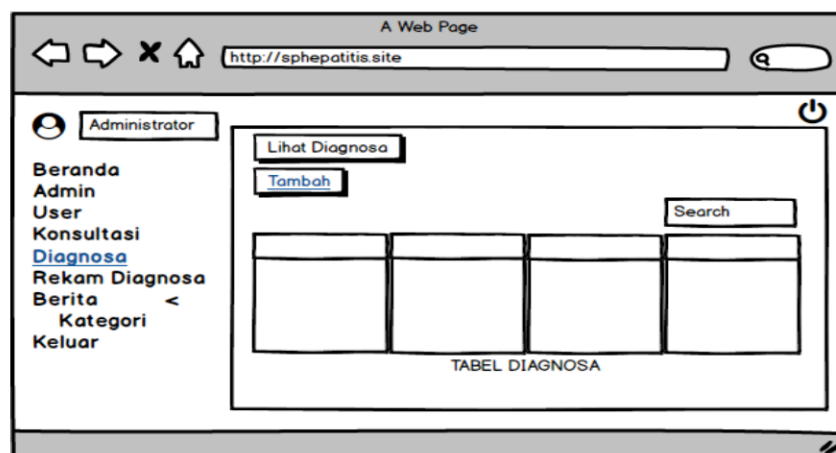
Tampilan perancangan dari halaman tambah kuisioner ditunjukkan pada gambar 4.56, halaman ini berisi *field* yang harus diisi sesuai dengan gejala penyakit yang berasal dari pakar. Halaman ini dibuat berdasarkan pada Tabel kuisioner pada Tabel 4.8.



Gambar 4.56 Rancangan halaman tambah kuisioner (*admin*)

17. Halaman diagnosis (*admin*)

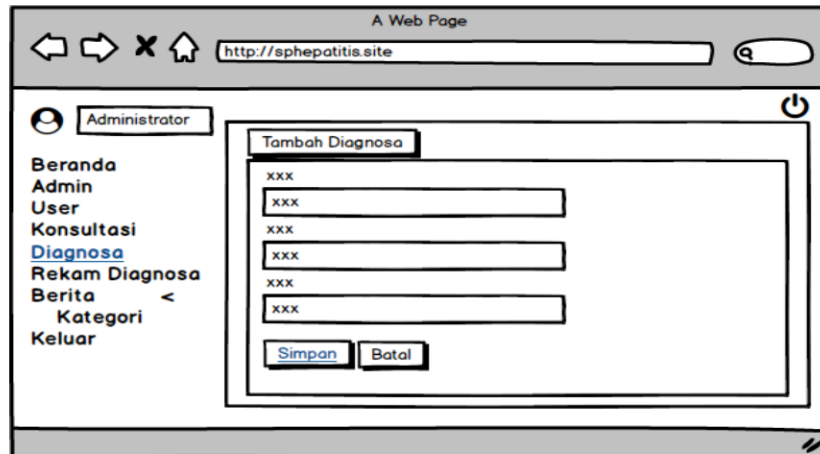
Tampilan perancangan dari halaman diagnosis ditunjukkan pada gambar 4.57, halaman ini berisi *data table* yang memuat jenis penyakit hepatitis. *Admin* dapat melakukan aksi tambah, edit, dan hapus. Halaman ini dibuat berdasarkan *activity diagram* diagnosis pada Gambar 4.16, *sequence diagram* diagnosis pada Gambar 4.35, serta data Tabel diagnosis pada Tabel 4.6.



Gambar 4.57 Rancangan halaman diagnosis (*admin*)

18. Halaman tambah diagnosis (*admin*)

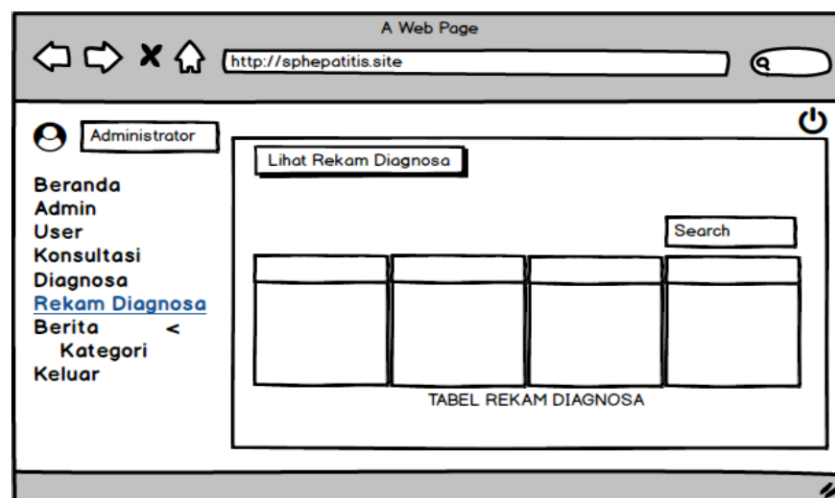
Gambar 4.58 merupakan desain halaman tambah diagnosis, pada halaman *admin* mengisikan nama penyakit, saran, dan pengobatan yang bersumber dari pakar. Halaman ini dibuat berdasarkan data Tabel diagnosis pada Tabel 4.6.



Gambar 4.58 Rancangan halaman tambah diagnosis (*admin*)

19. Halaman rekam diagnosis (*admin*)

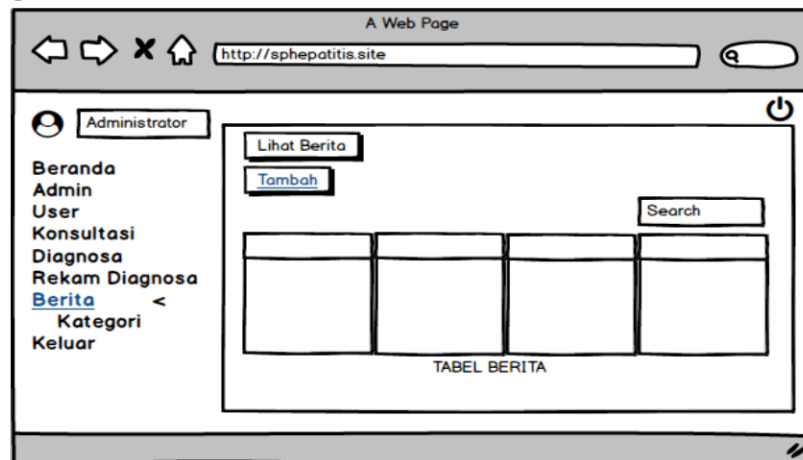
Gambar 4.59 merupakan desain halaman rekam diagnosis, pada halaman ini terdapat *data table* yang berisi riwayat rekam diagnosis yang telah dilakukan *user* sebelumnya. Halaman ini dibuat berdasarkan *activity diagram* rekam diagnosis pada Gambar 4.19, *sequence diagram* rekam diagnosis pada Gambar 4.38, serta data Tabel Rekam Diagnosis pada Tabel 4.9.



Gambar 4.59 Rancangan halaman rekam diagnosis (*admin*)

20. Halaman berita (*admin*)

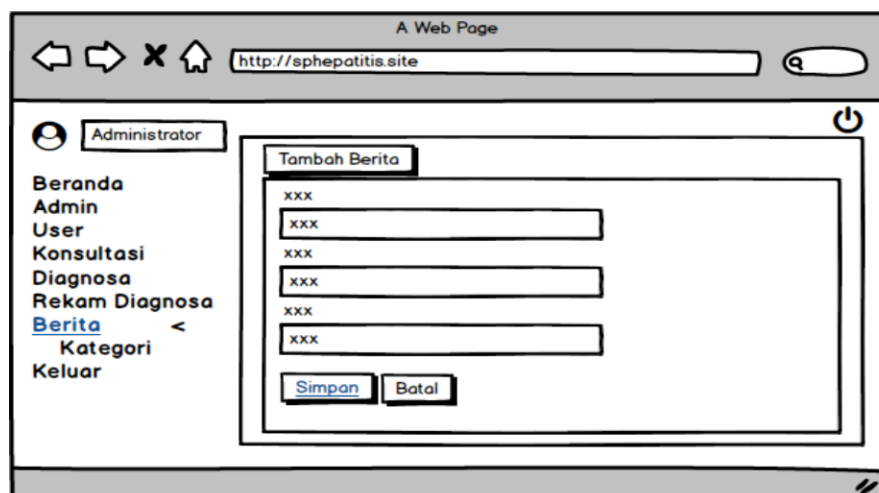
Tampilan perancangan dari halaman berita ditunjukkan pada gambar 4.60, pada halaman ini terdapat *data table* yang berisi daftar berita yang telah disubmit *admin*. *Admin* dapat melakukan aksi tambah, edit, hapus, dan *approve*. Halaman ini dibuat berdasarkan *activity diagram* berita pada Gambar 4.17, *sequence diagram* berita pada Gambar 4.36, serta data Tabel Berita pada Tabel 4.5.



Gambar 4.60 Rancangan halaman berita (*admin*)

21. Halaman tambah berita (*admin*)

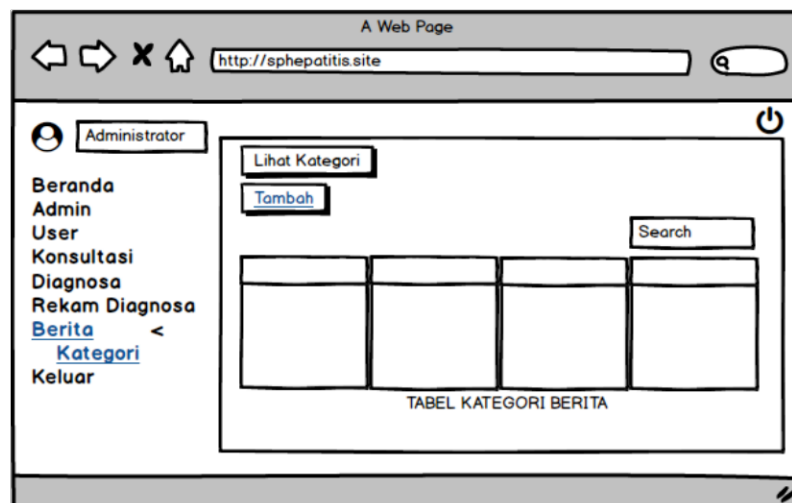
Tampilan perancangan dari halaman tambah berita ditunjukkan pada gambar 4.61, pada halaman ini *admin* menambahkan data berita berupa informasi kesehatan atau umum. Halaman ini dibuat berdasarkan data Tabel Berita pada Tabel 4.5.



Gambar 4.61 Rancangan halaman tambah berita (*admin*)

22. Halaman kategori berita (*admin*)

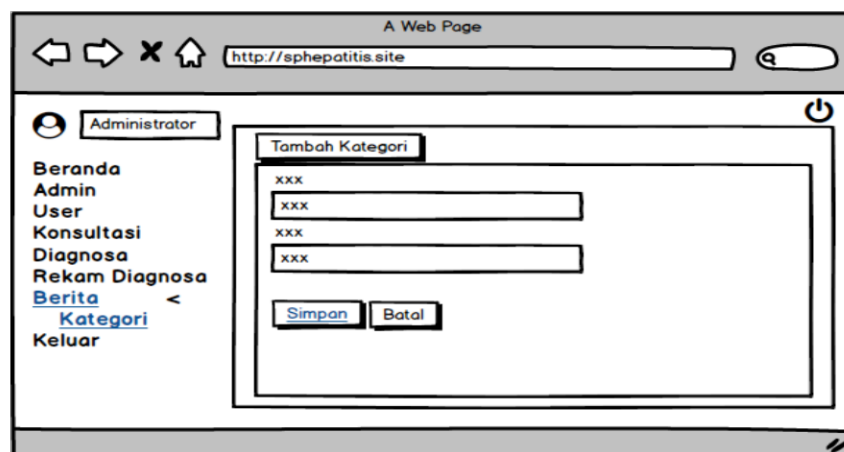
Gambar 4.62 merupakan desain halaman kategori berita, halaman ini berisi kategori berita yang dapat memudahkan pembagian jenis berita. *Admin* dapat melakukan aksi tambah, edit, dan hapus. Halaman ini dibuat berdasarkan *activity diagram* kategori berita pada Gambar 4.18, *sequence diagram* kategori berita pada Gambar 4.37, serta data Tabel Kategori pada Tabel 4.7.



Gambar 4.62 Rancangan halaman kategori berita (*admin*)

23. Halaman tambah kategori berita (*admin*)

Gambar 4.63 merupakan desain tampilan halaman tambah kategori berita, halaman ini berisi *field* yang harus diisi untuk menambahkan kategori berita. Halaman ini dibuat berdasarkan data Tabel Kategori pada Tabel 4.7.



Gambar 4.63 Rancangan halaman tambah kategori berita (*admin*)

4.5 Perancangan Sistem Pakar

4.5.1 Tabel Keputusan

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan juga merupakan matriks kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Walaupun kaidah secara langsung dapat dihasilkan dari tabel keputusan tetapi untuk menghasilkan kaidah yang efisien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan terlebih dahulu. Berikut tabel keputusan Sistem Pakar Penyakit Hepatitis.

Tabel 4.11 Tabel Keputusan

No	Kode Gejala	Gejala	Kode Penyakit		
			P1	P2	P3
1	G01	Flu disertai batuk pilek	*		
2	G02	Demam naik turun, suhu lebih dari 37.5 °C	*		
3	G03	Diare disertai lendir dan lebih dari 3 kali	*		
4	G04	Mual	*	*	
5	G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	*	*	*
6	G06	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	*	*	*
7	G07	Hilang nafsu makan	*	*	*
8	G08	Lemah	*		*
9	G09	Lesu	*	*	
10	G10	Pegal pada otot	*	*	*
11	G11	Kembung		*	*
12	G12	Muntah	*	*	*

13	G13	Air kencing gelap dikarenakan dehidrasi	*	*	*
----	-----	--	---	---	---

Keterangan:

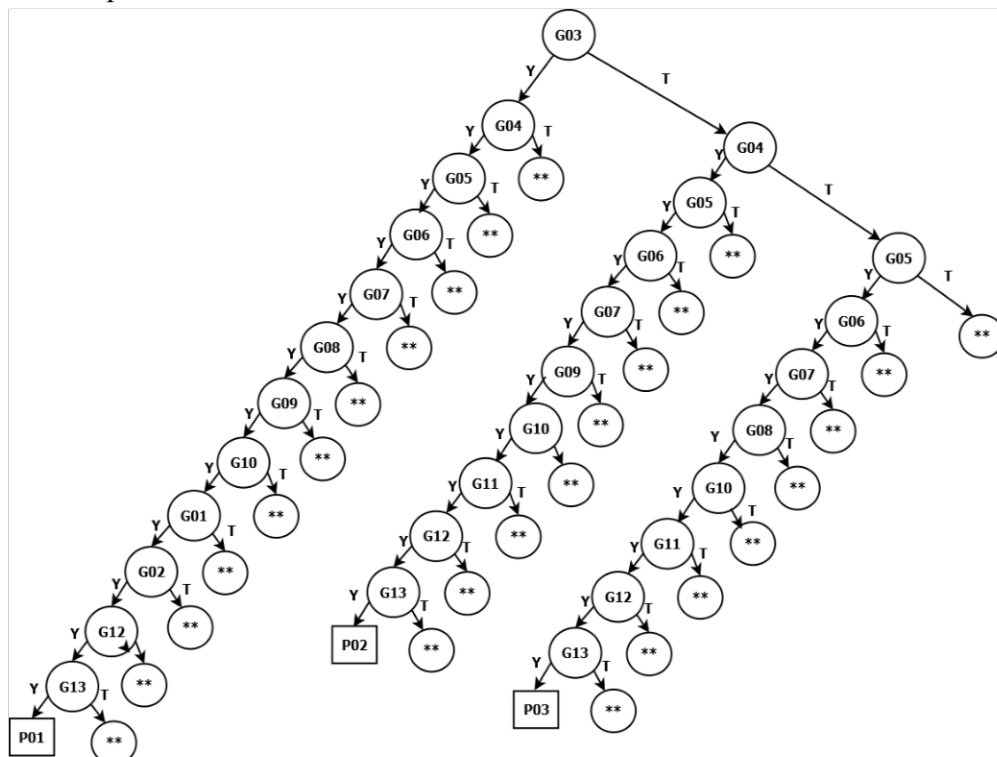
P1 = Hepatitis A

P2 = Hepatitis B

P3 = Hepatitis C

4.5.2 Pohon Keputusan

Diagram pohon keputusan merupakan suatu rancangan yang digunakan untuk membangun sebuah sistem pakar, di dalam diagram pohon keputusan tersebut akan dicari solusi hasil akhir dari setiap pemeriksaan. Diagram pohon keputusan akan mempermudah untuk menyusun basis pengetahuan dan aturan serta menentukan faktor kepastian dari setiap pelaksanaan diagnosa gejala pada penyakit hepatitis.



Gambar 4.64 Pohon Keputusan Sistem Pakar Diagnosis Hepatitis

Keterangan:

G1 – G13= Gejala

P01 – P03= Penyakit

* = Diluar prediksi sistem

4.5.3 Kaidah Produksi

Berikut ini kaidah produksi yang dapat dibuat berdasarkan tabel keputusan dan pohon keputusan yang telah dibuat.

1. **IF G03 AND G04 AND G05 AND G06 AND G07 AND G08 AND G09 AND G10 AND G01 AND G02 AND G12 AND G13 THEN P1.**
2. **IF G04 AND G05 AND G06 AND G07 AND G09 AND G10 AND G11 AND G12 AND G13 THEN P2.**
3. **IF G05 AND G06 AND G07 AND G08 AND G10 AND G11 AND G12 AND G13 THEN P3.**

BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil Penelitian

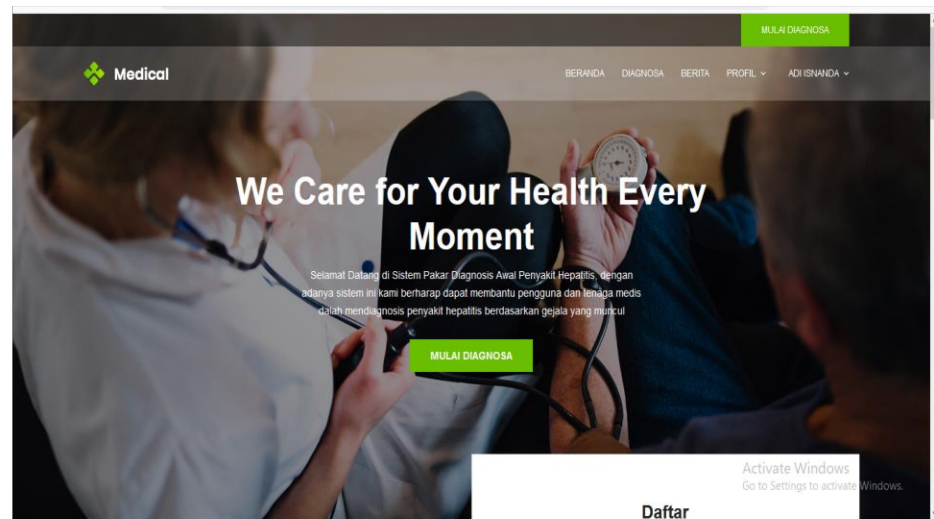
Berdasarkan analisis dan perancangan sistem maka pada penelitian ini dibangun sistem pakar diagnosis awal penyakit hepatitis. Penelitian tersebut menghasilkan sistem pakar dengan menu untuk *user* berupa menu diagnosis, login, berita, kontak, dan profil. Sedangkan untuk *Admin* tersedia beberapa menu untuk mengelola sumber daya sistem seperti menu user, admin, kuisioner, diagnosis, berita, dan laporan. Setelah tahap implementasi selesai, tahap selanjutnya pengujian *black box testing* dan validasi sistem.

5.1.1 Impelentasi Antarmuka

Implementasi antarmuka merupakan pemaparan mengenai tampilan sistem dan kegunaan fungsi dari setiap halaman yang ada. Untuk memperjelas bentuk dari implementasi antarmuka, berikut pemaparan dan fungsi dari setiap tampilan halaman yang telah dibuat.

a. Tampilan Halaman Beranda (*User*).

Halaman beranda adalah halaman yang pertama muncul saat *user* membuka sistem, halaman ini terdiri dari menu diagnosis, berita, profil, kontak, daftar akun, dan masuk (*login*). Halaman ini dibuat berdasarkan *mockup* halaman beranda *user* pada Gambar 4.41. Adapun tampilan halaman beranda dapat dilihat pada Gambar 5.1 berikut



Gambar 5.1 Tampilan Halaman beranda (*user*)

Serta halaman beranda *user* merupakan implemetasi dari potongan kode program (*controller*) yang dapat dilihat pada Gambar 5.2 berikut

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class HomeUsr extends CI_Controller {
    private $view = "frontend/v_homeusr/";
    private $redirect = "HomeUsr";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_homeusr');
    }

    public function index(){
        $newsterkini = $this->M_homeusr->newsterkini();
        $newshome=$this->M_homeusr->newshome();
        $data = array(
            'newsterkini' => $newsterkini,
            'newshome'=> $newshome|
        );
        $this->template->load('frontend/template',$this->view.'read', $data);
    }

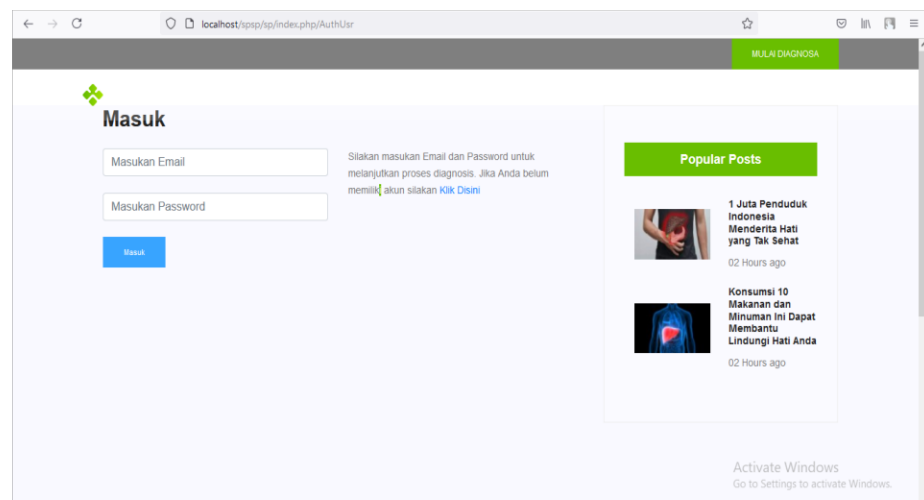
    public function readmore(){
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'row'=> $this->M_homeusr->readmore($this->uri->segment(3))->row_array()
        );
        $this->template->load('frontend/template',$this->view.'readmore', $data);
    }
}

```

Gambar 5.2 Controller beranda *user*

b. Tampilan Halaman *login (User)*.

Halaman *login* merupakan implementasi dari *mockup login user* pada Gambar 4.42, halaman ini disediakan oleh *admin* untuk mengisi *email* dan *password* agar *user* dapat mengakses fitur sistem yang diperuntukan bagi *user*. Tampilan halaman *login* dapat dilihat pada Gambar 5.3



Gambar 5.3 Tampilan Halaman *login (user)*

Serta, halaman *login user* merupakan implementasi dari potongan kode program (*controller*) yang dapat dilihat pada gambar 5.4 berikut:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class AuthUsr extends CI_Controller {
    private $view = "frontend/v_akunusr/";
    private $redirect = "AuthUsr";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_authusr');
        $this->load->model('M_homeusr');
    }

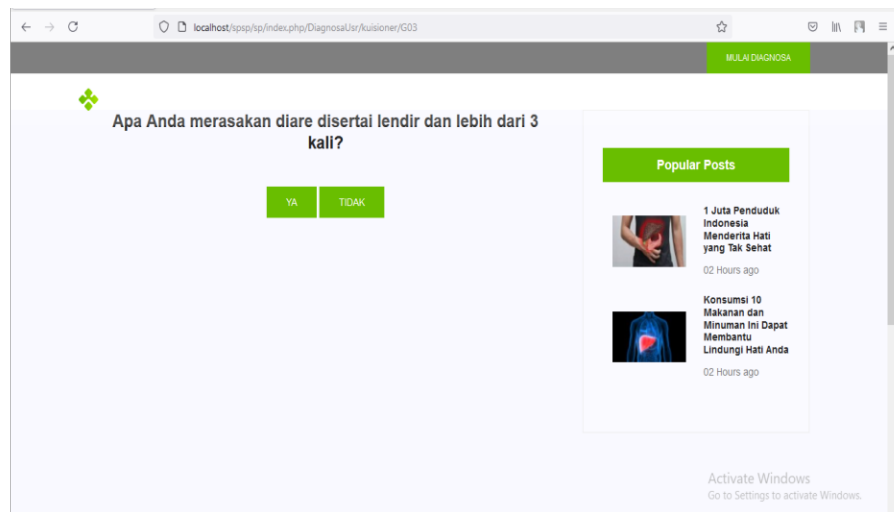
    public function index()
    {
        $this->session->sess_destroy();
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'read'=> ''
        );
        $this->template->load('frontend/template',$this->view.'login', $data);
    }

    public function login()
    {
        error_reporting(0);
        $id = $this->input->post('email_user');
        $pwd = md5($this->input->post('pswd_user'));
        $data = $this->M_authusr->cekLogin('user','email_user',$id);
        if($data['pswd_user'] == $pwd AND $data['email_user'] == $id){
            $array = array(
                'id_user' => $data['id_user'],
                'email_user' => $data['email_user'],
            );
        }
    }
}
```

Gambar 5.4 *Controller* halaman *login user*

c. Tampilan Halaman Diagnosis (*User*).

Tampilan menu diagnosis merupakan halaman yang disediakan untuk *user* yang telah *login* untuk memulai proses diagnosis awal penyakit hepatitis berdasarkan gejala-gejala yang dirasakan oleh *user*. Halaman ini dibuat berdasarkan *mockup* halaman diagnosis *user* pada Gambar 4.44, tampilan halaman diagnosis dapat dilihat pada Gambar 5.5



Gambar 5.5 Tampilan halaman diagnosis

Serta halaman diagnosis *user* merupakan implementasi dari potongan kode program (*controller* yang dapat dilihat pada Gambar 5.6 berikut:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class DiagnosaUsr extends CI_Controller {
    private $view = "frontend/v_diagnosa/";
    private $redirect = "DiagnosaUsr";

    public function __construct(){
        parent::__construct();
        //Load model
        //$this->load->model('M_kuisisionerusr');
        $this->load->model('M_diagnosausr');
        $this->load->model('M_homeusr');
        IsUser();
    }

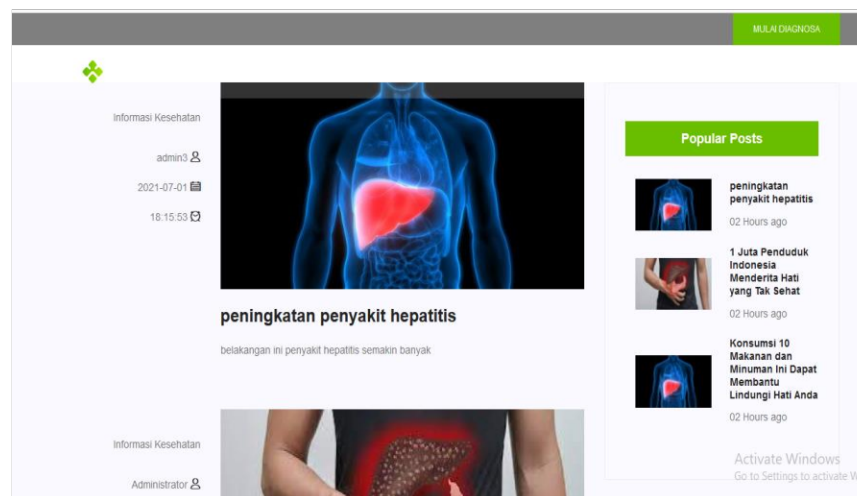
    public function kuisisioner(){
        $kd = $this->uri->segment(3);
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'analisa' => $this->M_diagnosausr->kuisisioner($kd)
        );
        $this->template->load('frontend/template',$this->view.'kuisisioner', $data);
    }

    public function hasildiagnosa(){
        $kd = $this->uri->segment(3);
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'analisa' => $this->M_diagnosausr->hasildiagnosa($kd)
        );
        $this->template->load('frontend/template',$this->view.'hasildiagnosa', $data);
    }
}
```

Gambar 5.6 *Controller* halaman diagnosis *user*

d. Tampilan Halaman Berita (*User*).

Tampilan halaman berita merupakan implementasi dari *mockup* berita *user* pada Gambar 4.45, tampilan ini dapat dilihat oleh *user* meskipun tanpa melakukan *login*, sebagai bahan bacaan terdapat kategori informasi kesehatan dan umum yang dapat dibaca oleh *user*. Tampilan halaman berita dapat dilihat dapat pada Gambar 5.7



Gambar 5.7 Tampilan halaman berita *user*

Serta halaman berita *user* merupakan implementasi dari potongan kode program (*controller*) yang dapat dilihat pada Gambar 5.8 berikut:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class BeritaUsr extends CI_Controller {
    private $view = "frontend/v_berita/";
    private $redirect = "NewsUsr";
    //Paging
    private $per_page = 4;

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_homeusr');
        $this->load->model('M_beritausr');
    }

    public function index(){

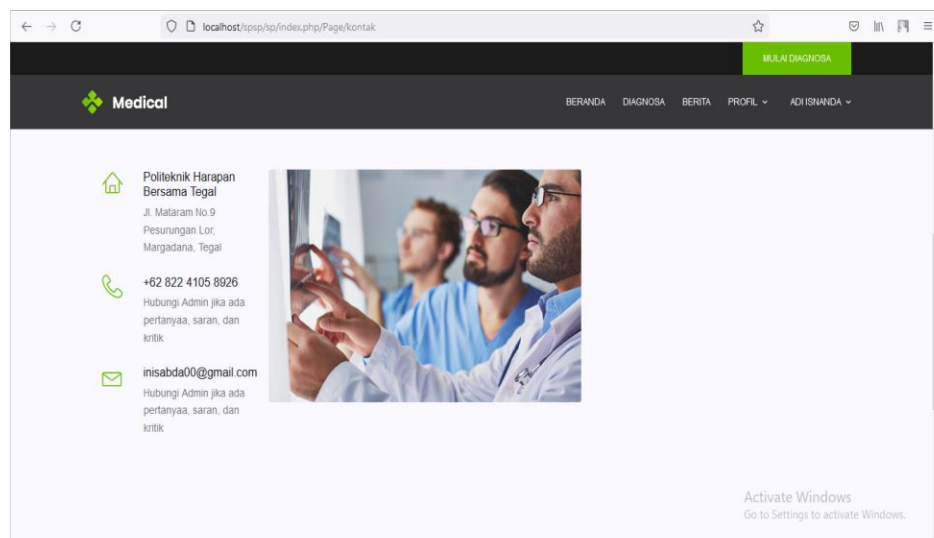
        $page= $this->uri->segment(3) ? $this->uri->segment(3) : 0;
        $newsusr=$this->M_beritausr->newsusr();
        $newsusr=$this->M_beritausr->GetDataPaging($this->per_page, $page);
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini'=> $newsterkini,
            'newsusr'=> $newsusr,
            'paging' => Paging($this->per_page,$this->M_beritausr->GetCount())
        );
        $this->template->load('frontend/template',$this->view.'read', $data);
    }

    public function readmore(){
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'row'=> $this->M_beritausr->readmore($this->uri->segment(3))->row_array()
        );
    }
}
```

Gambar 5.8 *Controller* halaman berita *user*

d. Tampilan Halaman Kontak (*User*).

Halaman ini berisi kontak informasi yang dapat dihubungi user ketika ada pertanyaan, saran, atau kritik yang berhubungan dengan sistem. Halaman ini dibuat berdasarkan *mokcup* kontak *user* pada Gambar 4.49, tampilan halaman kontak dapat dilihat pada Gambar 5.9



Gambar 5.9 Tampilan halaman kontak (*user*)

Serta halaman kontak *user* merupakan implementasi dari potongan kode program (*view*) yang dapat dilihat pada Gambar 5.10 berikut:

```
<!-- Start contact-page Area -->
<section class="contact-page-area section-gap">
  <div class="container">
    <div class="row">
      <div class="map-wrap" style="width:100%; height: 445px;" id="map"></div>
      <div class="col-lg-4 d-flex flex-column address-wrap">
        <div class="single-contact-address d-flex flex-row">
          <div class="icon">
            <span class="lnr lnr-home"></span>
          </div>
          <div class="contact-details">
            <h5>Politeknik Harapan Bersama Tegal</h5>
            <p>
              Jl. Mataram No.9 Pesurungan Lor, Margadana, Tegal
            </p>
          </div>
        </div>
        <div class="single-contact-address d-flex flex-row">
          <div class="icon">
            <span class="lnr lnr-phone-handset"></span>
          </div>
          <div class="contact-details">
            <h5>+62 822 4105 8926</h5>
            <p>Hubungi Admin jika ada pertanyaa,| saran, dan kritik</p>
          </div>
        </div>
        <div class="single-contact-address d-flex flex-row">
          <div class="icon">
            <span class="lnr lnr-envelope"></span>
          </div>
          <div class="contact-details">
            <h5>inisabda00@gmail.com</h5>
            <p>Hubungi Admin jika ada pertanyaa, saran, dan kritik</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

Gambar 5.10 *View* halaman kontak *user*

e. Tampilan Halaman Profil (*User*).

Halaman profil merupakan implementasi dari *mockup* halaman profil *user* pada Gambar 4.46, terdapat penjelasan mengenai profil sistem pakar penyakit hepatitis. *User* dapat melihat cara penggunaan serta fungsi buatnya sistem pakar ini. Tampilan halaman profil dapat dilihat pada Gambar 5.11 berikut



Gambar 5.11 Tampilan halaman profil (*user*)

Serta halaman profil *user* merupakan implementasi dari potongan kode program (*view*) pada Gambar 5.12 berikut

```

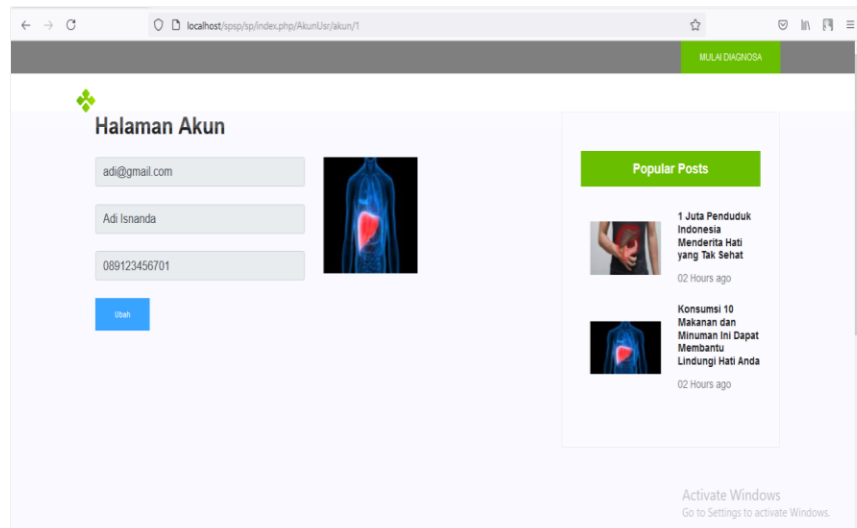
<div class="whole-wrap">
  <div class="container">
    <div class="section-top-border">
      <h3 class="mb-30"> Tentang Sistem Pakar Diagnosis Awal Hepatitis</h3>
      <div class="row">
        <div class="col-md-6">
          
          </div>
          <div class="col-md-6 mt-sm-20 left-align-p">
            <p> Dengan memanfaatkan perkembangan Teknologi, kami ikut membantu para pengguna untuk dapat melakukan diagnosis awal penyakit hepatitis. Dengan adanya sistem <p> <p> Menggunakan penggabungan Algoritma Forward Chaining dan Certainty Factor, diharapkan dapat melakukan proses diagnosis dengan tepat sesuai dengan sumber pe </div>
          </div>
        </div>
        <div class="section-top-border text-right">
          <h3 class="mb-30">Langkah Penggunaan Sistem Pakar Diagnosis Awal Hepatitis </h3>
          <div class="row">
            <div class="col-md-6">
              <p> <p> Untuk dapat melakukan proses diagnosis, pengguna diharuskan untuk memiliki akun terlebih dahulu agar mempermudah rekam diagnosis pengguna. Berikut adalah <p> <p> 1. Beranda akan tampil pertama saat Anda membuka sistem ini, untuk memulai diagnosis silahkan untuk membuat akun terlebih dahulu. </p> <p> 2. Pilih Menu Masuk selanjutnya pilih Menu Daftar Akun, silakan isi data pribadi Anda untuk keperluan proses Login. </p> <p> 3. Setelah Login dengan akun yang Anda buat, Anda merubah informasi pribadi dan Password Anda pada Menu Akun Saya. </p> <p> 4. Masuk ke proses diagnosis, silakan pilih Menu Mulai Diagnosis. </p> <p> 5. Pada Halaman Diagnosis, Anda akan ditampilkan pertanyaan berupa gejala gejala. Mohon isi sesuai gejala yang Anda rasakan. </p> <p> 6. Pada akhir proses diagnosis, akan ditampilkan hasil akhir jenis hepatitis sesuai dengan gejala gejala yang Anda masukan sebelumnya. </p> <p> 7. Pilih Menu Rekam untuk menyimpan Hasil Diagnosis Anda untuk keperluan kedepannya jika dibutuhkan. </p> </div>
            <div class="col-md-6">
              
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Gambar 5.12 *View* halaman profil *user*

f. Tampilan Halaman Akun Saya (*User*).

Halaman akun saya dibuat berdasarkan *mockup* halaman akun saya pada Gambar 4.48, terdapat data yang digunakan *user* untuk daftar akun serta merubah data pribadi. Tampilan halaman akun saya dapat dilihat pada Gambar 5.13 berikut.



Gambar 5.13 Tampilan halaman akun saya (*user*)

Serta merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.14 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class AkunUsr extends CI_Controller {
    private $view = "frontend/v_akunusr/";
    private $redirect = "AkunUsr";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_akunusr');
        $this->load->model('M_homeusr');
    }

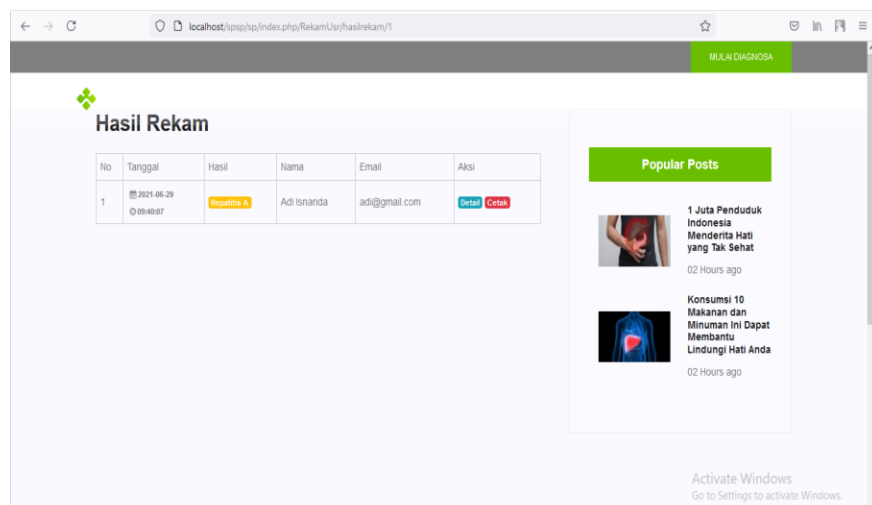
    public function index(){
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'read'=> ''
        );
        $this->template->load('frontend/template',$this->view.'daftar', $data);
    }

    public function save(){
        $pswd = $this->input->post('pswd_user');
        $kpswd = $this->input->post('kpswd_user');
        $email_user = $this->input->post('email_user');
        if($pswd != $kpswd){
            $this->session->set_flashdata('message_false','Password tidak sama');
            redirect('AkunUsr','refresh');
        }
        $where = array(
            'email_user' => $email_user
        );
        $cek = $this->M_akunusr->cekSama("user",$where)->num_rows();
        if($cek >= 1) {
            $this->session->set_flashdata('message_false','Email telah digunakan');
```

Gambar 5.14 *Controller* halaman akun saya (*user*)

g. Tampilan Halaman Hasil Diagnosis (*User*).

Pada halaman hasil diagnosis, terdapat data rekam diagnosis *user*. *User* dapat melihat detail diagnosis seperti jenis hepatitis, serta *user* mencetak hasil diagnosis. Halaman ini dibuat berdasarkan dari *mockup* hasil diagnosis pada Gambar 4.47, tampilan halaman hasil diagnosis dapat dilihat pada gambar 5.15 berikut.



Gambar 5.15 Tampilan halaman hasil diagnosis (*user*)

Serta halaman hasil diagnosis merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.16 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class RekamUsr extends CI_Controller {
    private $view = "frontend/v_rekam/";
    private $redirect = "RekamUsr";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_rekamusr');
        $this->load->model('M_homeusr');
        IsUser();
        $this->load->helper(array('url','download'));
    }

    public function download_diagnosa1(){
        force_download('assets/download/perhitungandiagnosa1.pdf',NULL);
    }

    public function download_diagnosa2(){
        force_download('assets/download/perhitungandiagnosa2.pdf',NULL);
    }

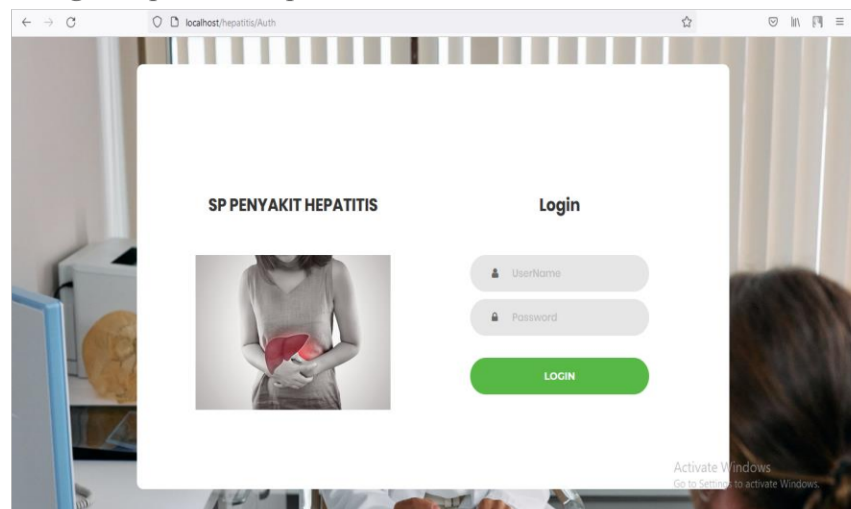
    public function download_diagnosa3(){
        force_download('assets/download/perhitungandiagnosa3.pdf',NULL);
    }

    public function hasilrekam(){
        $newsterkini = $this->M_homeusr->newsterkini();
        $data = array(
            'newsterkini' => $newsterkini,
            'data' => $this->M_rekamusr->hasilrekam($this->session->userdata('id_user'))
        );
        $this->template->load('frontend/template',$this->view.'read', $data);
    }
}
```

Gambar 5.16 *Controller* halaman hasil diagnosis (*user*)

h. Tampilan Halaman *Login (Admin)*

Halaman ini dibuat berdasarkan *mockup* halaman *login admin* pada Gambar 4.50, yang tampil ketika mengakses *admin*, terdapat *username* dan *login* yang harus diisi agar bisa masuk ke dalam sistem dan mengelola sumber daya. Tampilan halaman *login* dapat dilihat pada Gambar 5.17 berikut.



Gambar 5.17 Tampilan halaman *login (Admin)*

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.18 berikut

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Auth extends CI_Controller {

    private $redirect      = "Auth";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_auth');
    }

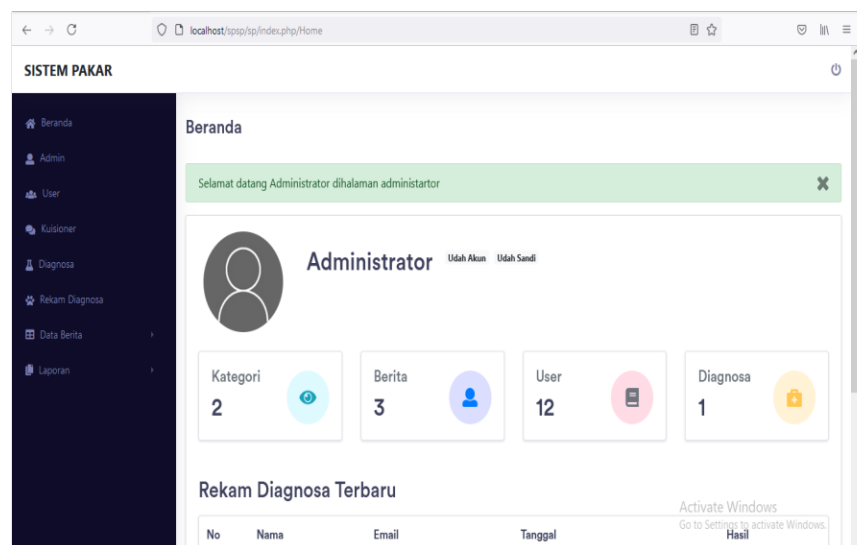
    public function index()
    {
        $this->session->sess_destroy();
        $data = array(
            'login' => ''
        );
        $this->load->view('backend/login', $data);
    }

    public function login()
    {
        $kd      = $this->input->post('kd_admin');
        $pswd    = md5($this->input->post('pswd_admin'));
        $data    = $this->M_auth->CekLogin('admin',$kd,$pswd);
        //jika login
        if($data>0)
        {
            $array = array(
                'kd_admin'      => $data['kd_admin'],
                'nama_admin'    => $data['nama_admin'],
                'img_admin'     => $data['img_admin'],
                'IsAdmin'       => 1
            );
        }
    }
}
```

Gambar 5.18 *Controller* halaman *login (Admin)*

i. Tampilan Halaman Beranda (*Admin*).

Halaman beranda merupakan implementasi dari *mockup* halaman beranda *admin* pada Gambar 4.50, halaman ini berisi informasi singkat mengenai sistem pakar seperti hasil diagnosis, jumlah *user*, dan berita. Tampilan halaman beranda dapat dilihat pada gambar 5.19 berikut.



Gambar 5.19 Tampilan halaman beranda (*Admin*)

Serta halaman ini dibuat berdasarkan dari potongan kode program (*controller*) dari Gambar 5.20 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Home extends CI_Controller {
    private $view = "backend/v_home/";
    private $redirect = "Home";

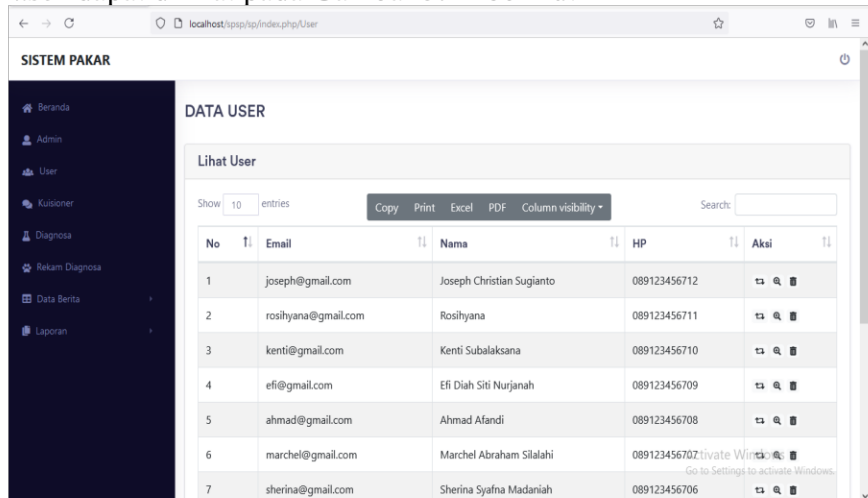
    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_home');
        IsAdmin();
    }

    public function index(){
        $user= $this->session->userdata('kd_admin');
        $rekam = $this->M_home->rekam();
        $data = array(
            'judul' => "Beranda",
            'sub' => "Halaman Beranda",
            'total_rekam' => $this->M_home->total('rekam'),
            'total_user' => $this->M_home->total('user'),
            'total_kategori' => $this->M_home->total('kategori'),
            'total_berita' => $this->M_home->total('berita'),
            'user'=> $this->M_home->user($user),
            'rekam'=> $rekam
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }
}
```

Gambar 5.20 *Controller* halaman beranda (*admin*)

j. Tampilan Halaman *User (Admin)*.

Tampilan halaman data *user*, terdapat *data table* yang memuat informasi mengenai *user* seperti nama, *email*, dan nomor hp. *Admin* hanya dapat melihat detail data *user* dan menghapusnya. Halaman ini dibuat berdasarkan *mockup* halaman *user (admin)* pada Gambar 4.54. Tampilan halaman *user* dapat dilihat pada Gambar 5.21 berikut



No	Email	Nama	HP	Aksi
1	joseph@gmail.com	Joseph Christian Sugianto	089123456712	[Edit] [Delete]
2	rosihyana@gmail.com	Rosihyana	089123456711	[Edit] [Delete]
3	kenti@gmail.com	Kenti Subalaksana	089123456710	[Edit] [Delete]
4	efi@gmail.com	Efi Diah Siti Nurjanah	089123456709	[Edit] [Delete]
5	ahmad@gmail.com	Ahmad Afandi	089123456708	[Edit] [Delete]
6	marchel@gmail.com	Marchel Abraham Silalahi	089123456707	[Edit] [Delete]
7	sherina@gmail.com	Sherina Syafna Madaniah	089123456706	[Edit] [Delete]

Gambar 5.21 Tampilan halaman *user (Admin)*

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.22 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class User extends CI_Controller {
    private $view = "backend/v_user/";
    private $redirect = "User";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_user');
        Isadmin();
    }

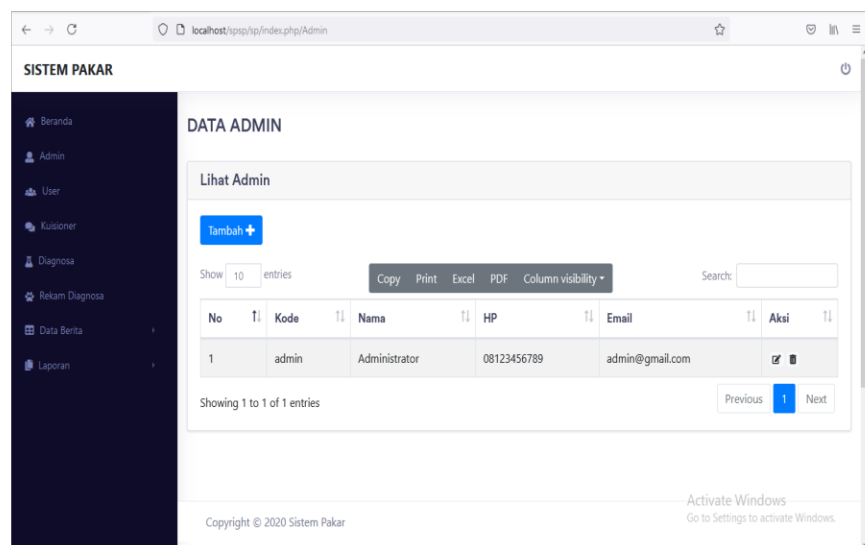
    public function index(){
        if ($this->input->get('search')) {
            $read = $this->M_user->search($this->input->get('search'));
        }else{
            $read = $this->M_user->GetAll();
        }
        $data = array(
            'judul' => "DATA USER",
            'sub' => "Lihat User",
            'read'=> $read
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

    public function cetak(){
        if ($this->input->get('search')) {
            $read = $this->M_user->search($this->input->get('search'));
        }else{
            $read = $this->M_user->GetAll();
        }
        $data = array(
            'judul' => "DATA USER",
            'sub' => "Lihat User",
```

Gambar 5.22 *Controller* halaman *user (admin)*

k. Tampilan Halaman *Admin (Admin)*.

Tampilan halaman data *admin*, halaman ini berisi *data table* yang memuat informasi *admin* yang dapat mengelola sumber daya sistem pakar. *Admin* dapat melakukan tambah, edit dan hapus data. Halaman ini merupakan implementasi dari *mockpup* halaman admin pada Gambar 4.52. Adapun halaman *admin* dapat dilihat pada Gambar 5.23 berikut



Gambar 5.23 Tampilan halaman *admin (Admin)*

Serta halaman ini dibuat berdasarkan dari potongan kode program (*controller*) pada Gambar 5.24 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Admin extends CI_Controller {
    private $view = "backend/v_admin/";
    private $redirect = "Admin";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_admin');
        isAdmin();
    }

    public function index(){
        if ($this->input->get('search')) {
            $read = $this->M_admin->search($this->input->get('search'));
        }else{
            $read = $this->M_admin->GetAll();
        }
        $data = array(
            'judul' => "DATA ADMIN",
            'sub' => "Lihat Admin",
            'read'=> $read
        );

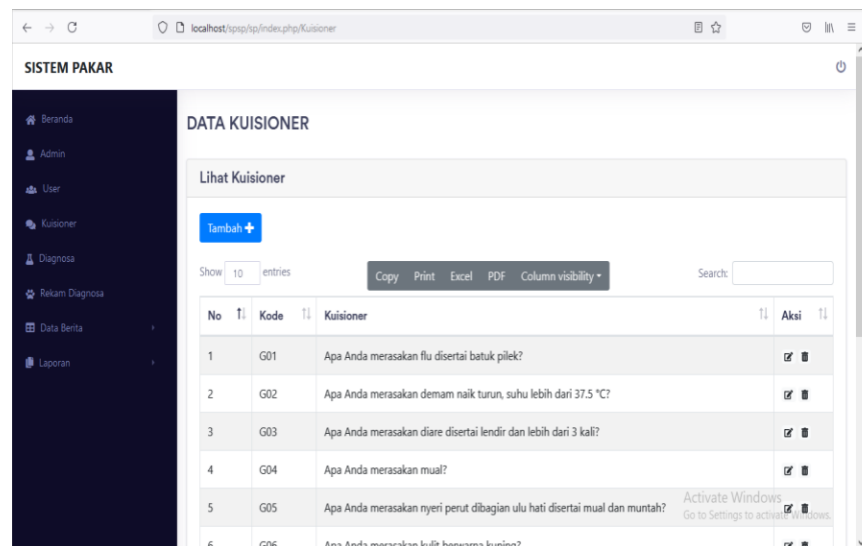
        $this->template->load('backend/template',$this->view.'read', $data);
    }

    public function create(){
        $data = array(
            'judul' => "DATA ADMIN",
            'sub' => "Tambah Admin",
            'create' => ''
        );
        $this->template->load('backend/template',$this->view.'create', $data);
    }
}
```

Gambar 5.24 *Controller* halaman *admin (admin)*

1. Tampilan Halaman Kuisisioner (*Admin*).

Tampilan data kuisisioner berisi halaman yang disediakan untuk menampilkan *data table* kuisisioner, pada halaman ini *admin* dapat melakukan aksi tambah, edit dan hapus. Halaman ini dibuat berdasarkan *mockup* halaman kuisisioner pada Gambar 4.55, tampilan halaman data kuisisioner dapat dilihat pada gambar 5.25 berikut.



Gambar 5.25 Tampilan Halaman kuisisioner (*Admin*)

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.26 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Kuisisioner extends CI_Controller {
    private $view = "backend/v_kuisisioner/";
    private $redirect = "Kuisisioner";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_kuisisioner');
        //Cek Login
        IsAdmin();
    }

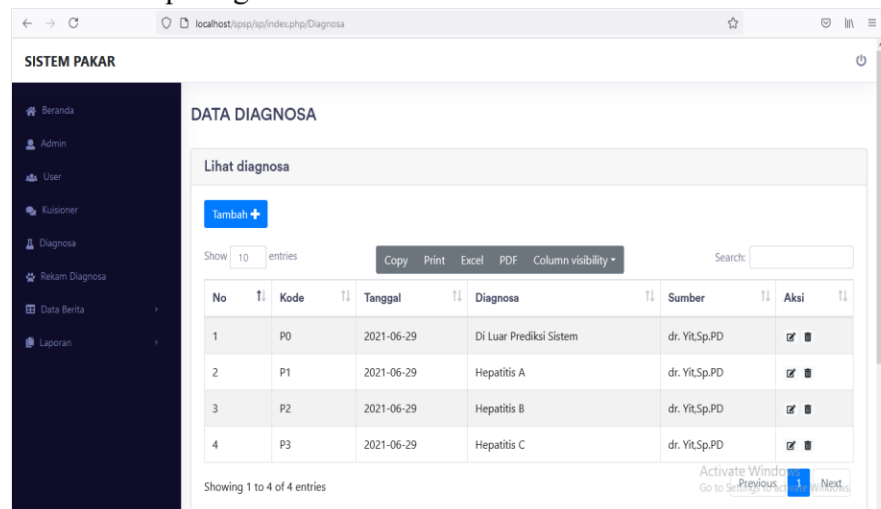
    public function index(){
        if ($this->input->get('search')){
            $read = $this->M_kuisisioner->search($this->input->get('search'));
        }
        else{
            $read = $this->M_kuisisioner->GetAll();
        }
        $data = array(
            'judul' => "DATA KUISISIONER",
            'sub' => "Lihat Kuisisioner",
            'read'=> $read
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

    public function create(){
        $data = array(
            'judul' => "DATA KUISISIONER",
            'sub' => "Tambah Kuisisioner",
            'create' => ''
        );
        $this->template->load('backend/template',$this->view.'create', $data);
    }
}
```

Gambar 5.26 *Controller* halaman kuisisioner (*admin*)

m. Tampilan Halaman Diagnosis (*Admin*).

Halaman diagnosis merupakan implementasi *mockup* halaman diagnosis *admin* pada Gambar 4.57, menjelaskan *data table* diagnosis, *admin* dapat melakukan aksi tambah, edit dan hapus data diagnosis. Tampilan halaman data diagnosis dapat dilihat pada gambar 5.27 berikut.



Gambar 5.27 Tampilan halaman diagnosis (*Admin*)

Serta halaman ini dibuat berdasarkan potongan kode program (*Controller*) pada Gambar 5.28 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Diagnosa extends CI_Controller {
    private $view = "backend/v_diagnosa/";
    private $redirect = "Diagnosa";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_diagnosa');
        //Cek Login
        IsAdmin();
    }

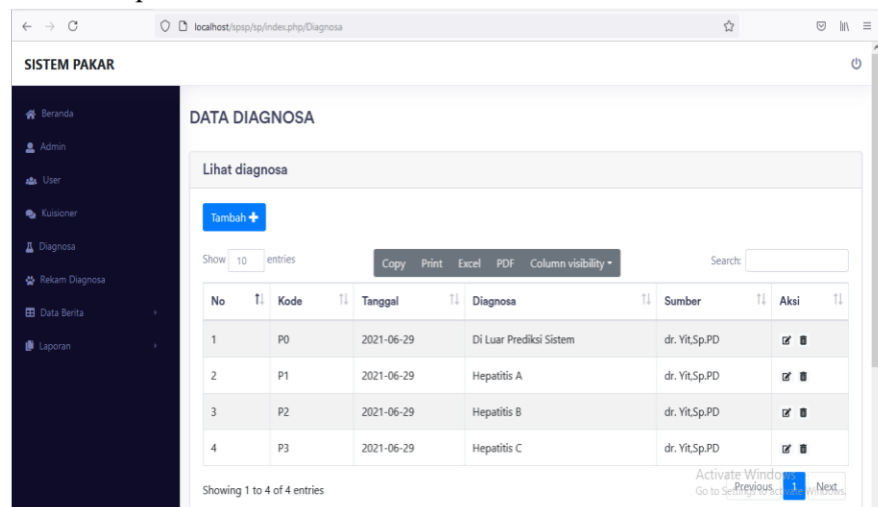
    public function index(){
        if ($this->input->get('search')){
            $read = $this->M_diagnosa->search($this->input->get('search'));
        }
        else{
            $read = $this->M_diagnosa->GetAll();
        }
        $data = array(
            'judul' => "DATA DIAGNOSA",
            'sub' => "Lihat diagnosa",
            'read'=> $read
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

    public function create(){
        $data = array(
            'judul' => "DATA diagnosa",
            'sub' => "Tambah diagnosa",
            'create' => ''
        );
        $this->template->load('backend/template',$this->view.'create', $data);
    }
}
```

Gambar 5.28 *Controller* halaman diagnosis (*admin*)

n. Tampilan Halaman Rekam Diagnosis (*Admin*).

Tampilan rekam diagnosis, halaman ini berisi *data table* rekam diagnosis yang telah dilakukan *user* sebelumnya. *Admin* hanya dapat melakukan aksi melihat detail dan menghapus data. Halaman ini dibuat berdasarkan *mockup* rekam diagnosis *admin* pada Gambar 4.59. Tampilan halaman rekam diagnosis dapat dilihat pada Gambar 5.29 berikut



Gambar 5.29 Tampilan halaman rekam diagnosis (*Admin*)

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.30 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Rekam extends CI_Controller {
    private $view = "backend/v_rekam/";
    private $redirect = "Rekam";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_rekam');
        IsAdmin();
    }

    public function index(){
        $q = $this->M_rekam->GetAll();
        $data = array(
            'judul' => "DATA REKAM DIAGNOSA",
            'sub' => "Lihat Rekam Diagnosa",
            'read'=> $q
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

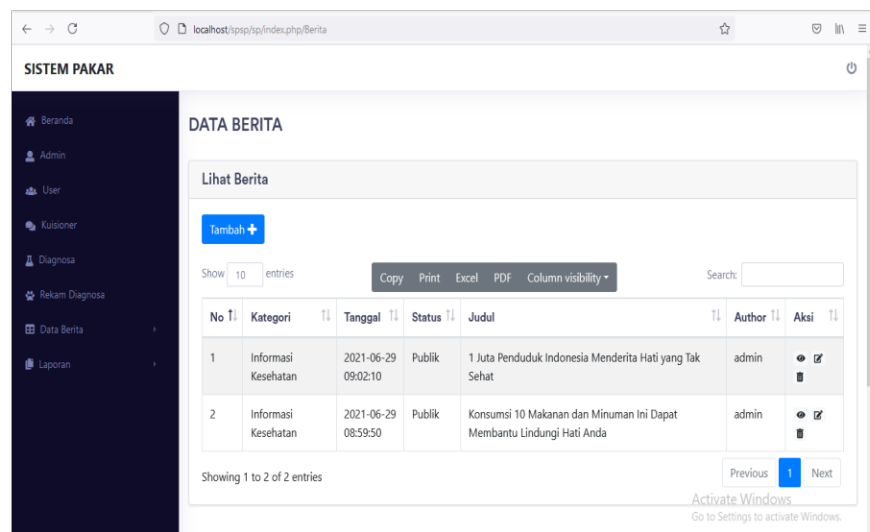
    public function cetak(){
        $q = $this->M_rekam->GetAll();
        $data = array(
            'judul' => "DATA REKAM DIAGNOSA",
            'sub' => "Lihat Rekam Diagnosa",
            'read'=> $q
        );
        $this->load->view($this->view.'cetak', $data);
    }

    public function detail(){
        $kd = $this->uri->segment(3);
        $data = array(
            'judul' => "DATA REKAM DIAGNOSA",
            'sub' => "Lihat Rekam Diagnosa",
            'read'=> $q
        );
    }
}
```

Gambar 5.30 *Controller* halaman rekam diagnosis (*admin*)

o. Tampilan Halaman Berita (*Admin*).

Pada gambar 5.31 menampilkan halaman berita, pada halaman ini berisi *data table* berita yang berisi informasi yang dapat dibaca *user*. *Admin* dapat melakukan aksi tambah, edit, hapus dan *approve* berita. Halaman ini dibuat berdasarkan *mockup* berita *admin* pada Gambar 4.60.



Gambar 5.31 Tampilan halaman berita (*Admin*)

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) pada Gambar 5.32 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Berita extends CI_Controller {
    private $view = "backend/v_berita/";
    private $redirect = "Berita";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_berita');
        $this->load->model('M_kategori');
        $this->load->library('upload');
        IsAdmin();
    }

    public function index(){
        $q = $this->M_berita->GetAll();
        $data = array(
            'judul' => "DATA BERITA",
            'sub' => "Lihat Berita",
            'read'=> $q
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

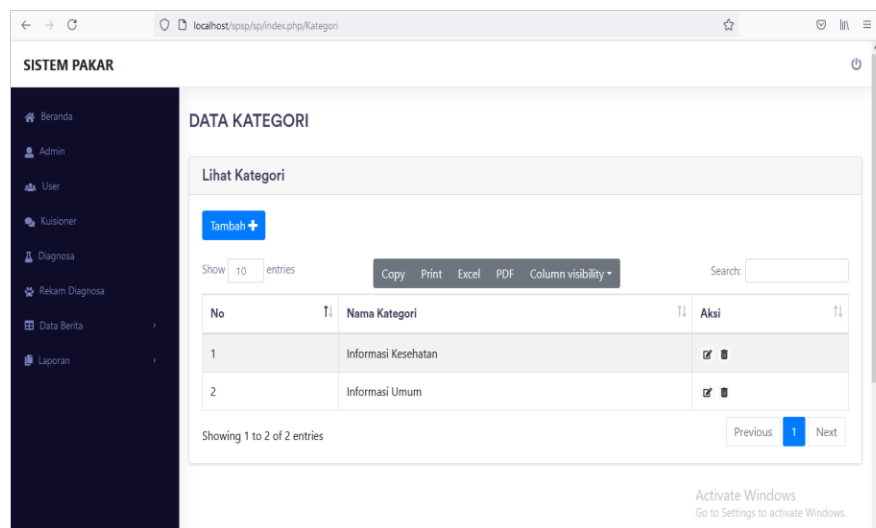
    public function cetak(){
        $q = $this->M_berita->GetAll();
        $data = array(
            'judul' => "DATA BERITA",
            'sub' => "Lihat Berita",
            'read'=> $q
        );
        $this->load->view($this->view.'cetak', $data);
    }

    public function create(){
```

Gambar 5.32 *Controller* halaman berita (*Admin*)

p. Tampilan Halaman Kategori Berita (*Admin*).

Pada gambar 5.33 menampilkan halaman kategori berita. Pada halaman ini terdapat pengelompokan kategori berita menjadi informasi kesehatan dan umum. *Admin* dapat melakukan aksi tambah, edit, dan hapus. Halaman ini dibuat berdasarkan *mockup* kategori berita pada Gambar 4.61.



Gambar 5.33 Tampilan halaman kategori berita (*Admin*)

Serta halaman ini merupakan implementasi dari potongan kode program (*controller*) dari Gambar 5.34 berikut

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Kategori extends CI_Controller {
    private $view = "backend/v_kategori/";
    private $redirect = "Kategori";

    public function __construct(){
        parent::__construct();
        //Load model
        $this->load->model('M_kategori');
        //Cek Login
        IsAdmin();
    }

    public function index(){
        if ($this->input->get('search')){
            $read = $this->M_kategori->search($this->input->get('search'));
        }
        else{
            $read = $this->M_kategori->GetAll();
        }
        $data = array(
            'judul' => "DATA KATEGORI",
            'sub' => "Lihat Kategori",
            'read'=> $read
        );
        $this->template->load('backend/template',$this->view.'read', $data);
    }

    public function create(){
        $data = array(
            'judul' => "DATA KATEGORI",
            'sub' => "Tambah Kategori",
            'create' => ''
        );
        $this->template->load('backend/template',$this->view.'create', $data);
    }
}
```

Gambar 5.34 *Controller* halaman kategori berita (*admin*)

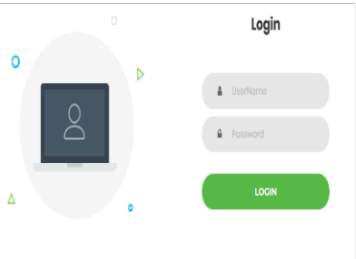
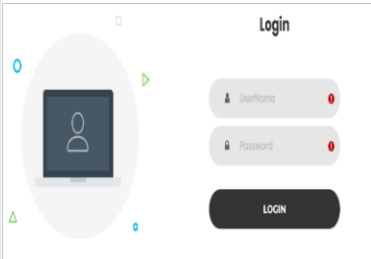
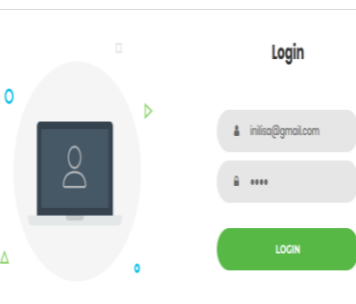
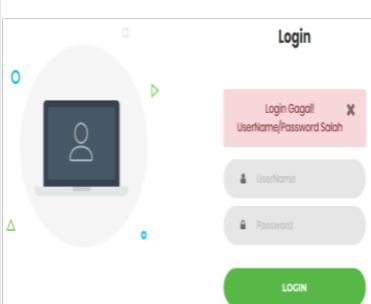

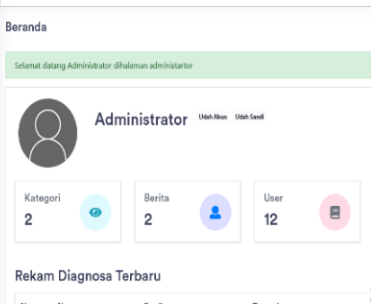
5.1.2 *Black Box Testing*

Black box testing dilakukan dengan mengecek fungsionalitas dan *output* dari sistem yang dibuat, yaitu pengujian terhadap halaman yang mempunyai masukan (*input*) dan keluaran (*output*) untuk mengetahui apakah suatu fungsi terjadi proses lalu lintas data sudah sesuai dengan yang diharapkan atau tidak.

a. Pengujian Halaman *Login Admin*

Pengujian halaman *login* merupakan pengujian yang dilakukan terhadap kode *user* dan *password*. Tampilan hasil pengujian halaman *login* dapat dilihat pada Tabel 5.1 berikut.

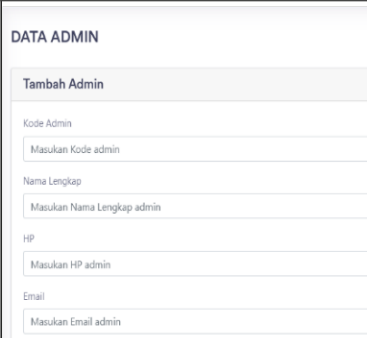
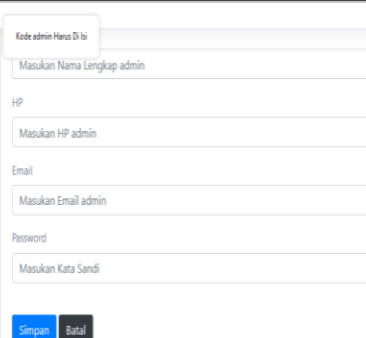
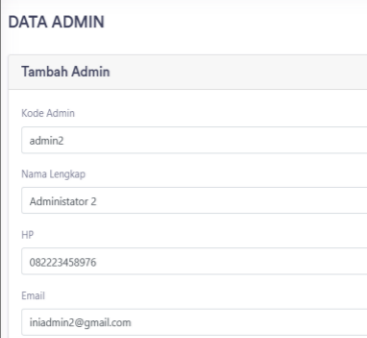
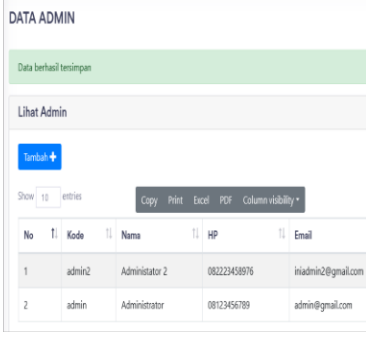
Tabel 5.1 Pengujian halaman login admin

<i>Field</i>	<i>Input</i>	<i>Output</i>	<i>Kete- rangan</i>
<i>Email & Password</i>	 <p><i>null</i></p>	 <p><i>error message</i></p>	Ditolak
<i>Email & Password</i>	 <p><i>Data!=database</i></p>	 <p><i>Error message</i></p>	Ditolak
<i>Email & Password</i>	 <p><i>Data=database</i></p>	 <p><i>Valid</i></p>	Diterima

b. Pengujian Halaman *Admin*

Pengujian halaman admin merupakan pengujian terhadap fungsionalitas halaman admin. Tampilan hasil pengujian halaman admin dapat dilihat pada Tabel 5.2 berikut.

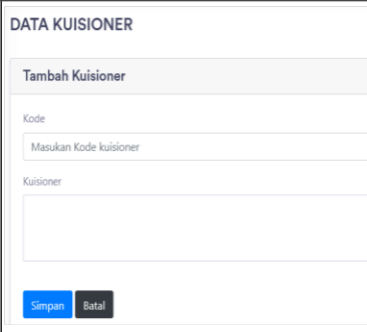
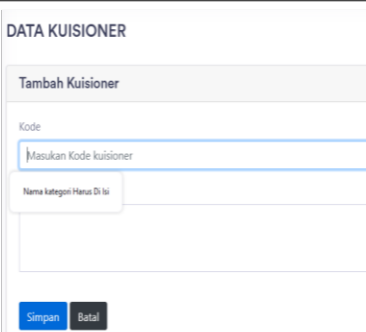
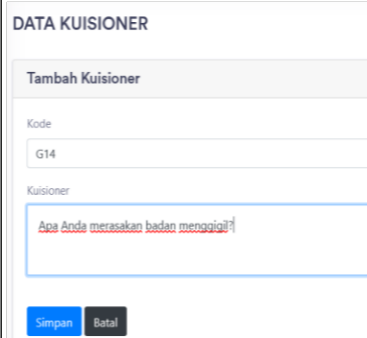
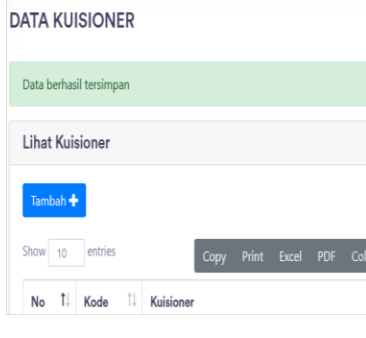
Tabel 5.2 Tampilan pengujian halaman admin

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan															
Kode admin, nama, hp, email, & password	 <p>DATA ADMIN</p> <p>Tambah Admin</p> <p>Kode Admin Masukan Kode admin</p> <p>Nama Lengkap Masukan Nama Lengkap admin</p> <p>HP Masukan HP admin</p> <p>Email Masukan Email admin</p> <p><i>nuu</i></p>	 <p>Kode admin Harus Di Isi</p> <p>Masukan Nama Lengkap admin</p> <p>HP Masukan HP admin</p> <p>Email Masukan Email admin</p> <p>Password Masukan Kata Sandi</p> <p>Simpan Batal</p> <p><i>error message</i></p>	Ditolak															
Kode admin, nama, hp, email, & password	 <p>DATA ADMIN</p> <p>Tambah Admin</p> <p>Kode Admin admin2</p> <p>Nama Lengkap Administrator 2</p> <p>HP 082223458976</p> <p>Email iniadmin2@gmail.com</p> <p><i>Data=database</i></p>	 <p>DATA ADMIN</p> <p>Data berhasil tersimpan</p> <p>Lihat Admin</p> <p>Tersulut</p> <p>Show 10 entries Copy Print Excel PDF Column visibility</p> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Nama</th> <th>HP</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>admin2</td> <td>Administrator 2</td> <td>082223458976</td> <td>iniadmin2@gmail.com</td> </tr> <tr> <td>2</td> <td>admin</td> <td>Administrator</td> <td>08123456789</td> <td>admin@gmail.com</td> </tr> </tbody> </table> <p><i>Valid</i></p>	No	Kode	Nama	HP	Email	1	admin2	Administrator 2	082223458976	iniadmin2@gmail.com	2	admin	Administrator	08123456789	admin@gmail.com	Diterima
No	Kode	Nama	HP	Email														
1	admin2	Administrator 2	082223458976	iniadmin2@gmail.com														
2	admin	Administrator	08123456789	admin@gmail.com														

c. Pengujian Halaman Kuisisioner (*Admin*)

Pengujian halaman kuisisioner merupakan pengujian terhadap fungsionalitas halaman kuisisioner. Tampilan hasil pengujian halaman kuisisioner dapat dilihat pada Tabel 5.3 berikut.


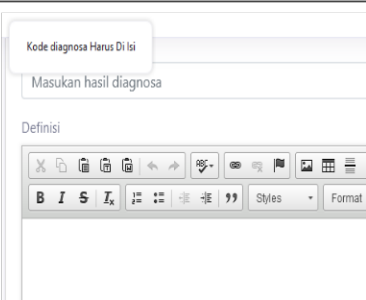

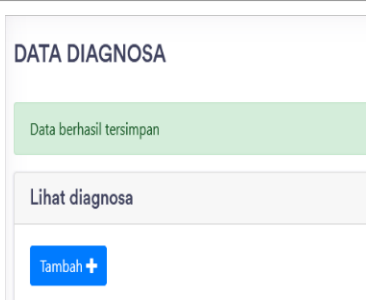
Tabel 5.3 Pengujian halaman kuisisioner (*admin*)

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan
Kode, kuisisioner	 <p style="text-align: center;"><i>null</i></p>	 <p style="text-align: center;"><i>error message</i></p>	Ditolak
Kode, kuisisioner	 <p style="text-align: center;"><i>Data=database</i></p>	 <p style="text-align: center;"><i>Valid</i></p>	Diterima

d. Pengujian Halaman Diagnosis (*Admin*)

Pengujian halaman diagnosis merupakan pengujian terhadap fungsionalitas halaman diagnosis. Tampilan hasil pengujian halaman diagnosis dapat dilihat pada Tabel 5.4 berikut.


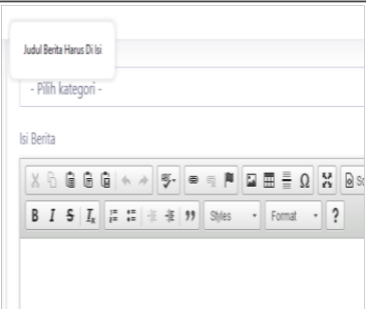


Tabel 5.4 Pengujian halaman diagnosis (admin)

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan
Kode, diagnosis, definisi, saran, tanggal, sumber	 <p>DATA diagnosa</p> <p>Tambah diagnosa</p> <p>Kode Masukan Kode diagnosa</p> <p>Diagnosa Masukan hasil diagnosa</p> <p>Definisi</p> <p><i>null</i></p>	 <p>Kode diagnosa Harus Di Isi</p> <p>Masukan hasil diagnosa</p> <p>Definisi</p> <p><i>error message</i></p>	Ditolak
Kode, diagnosis, definisi, saran, tanggal, sumber	 <p>DATA diagnosa</p> <p>Tambah diagnosa</p> <p>Kode P5</p> <p>Diagnosa PENYAKIT</p> <p><i>Data=database</i></p>	 <p>DATA DIAGNOSA</p> <p>Data berhasil tersimpan</p> <p>Lihat diagnosa</p> <p>Tambah +</p> <p><i>Valid</i></p>	Diterima

e. Pengujian Halaman Berita (*Admin*)

Pengujian halaman berita merupakan pengujian terhadap fungsionalitas halaman berita. Tampilan hasil pengujian halaman berita dapat dilihat pada Tabel 5.5 berikut.


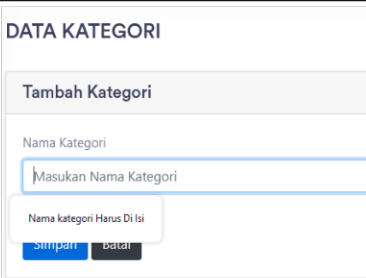


Tabel 5.5 Pengujian halaman berita (admin)

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan
Judul, kategori, isi berita, foto berita.	 <p>DATA BERITA</p> <p>Tambah Berita</p> <p>Judul Berita</p> <p>Masukan Judul Berita</p> <p>Kategori</p> <p>- Pilih kategori -</p>	 <p>Judul Berita Harus Di Isi</p> <p>- Pilih kategori -</p> <p>Isi Berita</p> <p>B I S I</p> <p>Styles - Format - ?</p>	Ditolak
	<i>null</i>	<i>error message</i>	
Judul, kategori, isi berita, foto berita.	 <p>DATA BERITA</p> <p>Tambah Berita</p> <p>Judul Berita</p> <p>headline hari ini</p> <p>Kategori</p> <p>Informasi Kesehatan</p>	 <p>DATA BERITA</p> <p>Data berhasil tersimpan</p> <p>Lihat Berita</p> <p>Tambah +</p>	Diterima
	<i>Data=database</i>	<i>Valid</i>	

f. Pengujian Halaman Kategori Berita (*Admin*)

Pengujian halaman kategori berita merupakan pengujian terhadap fungsionalitas halaman kategori berita. Tampilan hasil pengujian halaman kategori berita dapat dilihat pada Tabel 5.6 berikut.

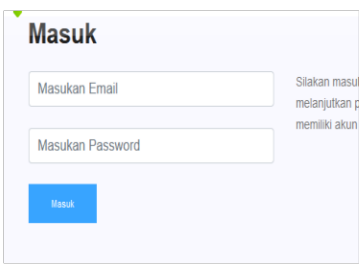
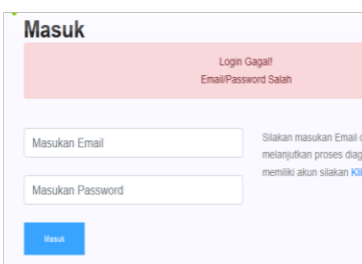
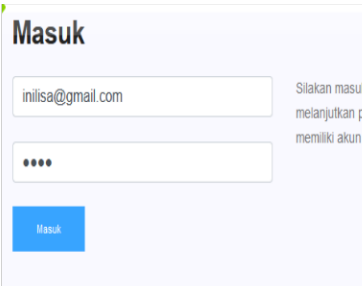
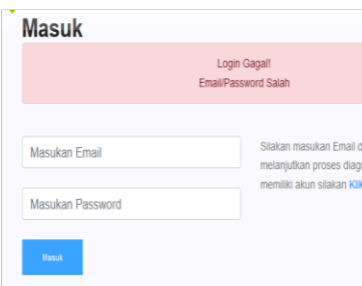
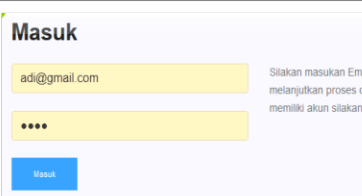

Tabel 5.6 Pengujian halaman kategori berita (admin)

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan
Nama kategori	 <p>DATA KATEGORI</p> <p>Tambah Kategori</p> <p>Nama Kategori</p> <p>Masukan Nama Kategori</p> <p>Simpan Batal</p> <p><i>null</i></p>	 <p>DATA KATEGORI</p> <p>Tambah Kategori</p> <p>Nama Kategori</p> <p>Masukan Nama Kategori</p> <p>Nama kategori Harus Di Isi</p> <p>Simpan Batal</p> <p><i>error message</i></p>	Ditolak
Nama kategori	 <p>DATA KATEGORI</p> <p>Tambah Kategori</p> <p>Nama Kategori</p> <p>Kategori Lainnya</p> <p>Simpan Batal</p> <p><i>Data=database</i></p>	 <p>DATA KATEGORI</p> <p>Data berhasil tersimpan</p> <p>Lihat Kategori</p> <p>Tambah +</p> <p><i>Valid</i></p>	Diterima

g. Pengujian halaman *login user*

Pengujian halaman *login* merupakan pengujian yang dilakukan terhadap kode *user* dan *password*. Tampilan hasil pengujian halaman *login* dapat dilihat pada Tabel 5.7 berikut.

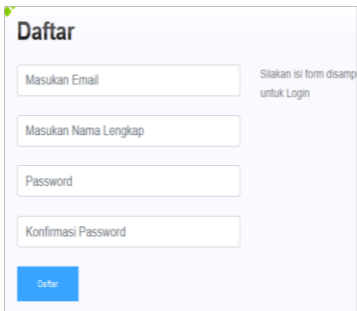
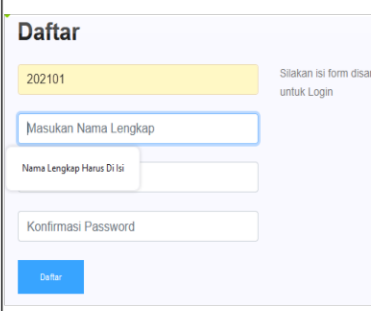
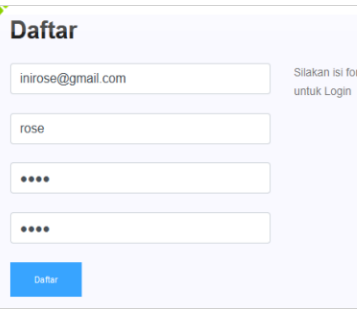
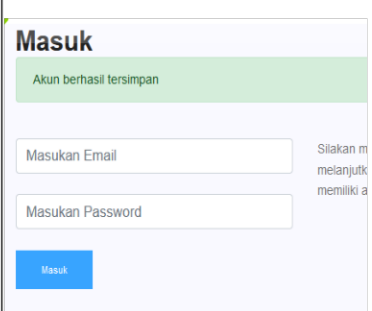
Tabel 5.7 Pengujian halaman login user

<i>Field</i>	<i>Input</i>	<i>Output</i>	Kete- rangan
<i>Email & Password</i>	 <p>Masuk</p> <p>Masukan Email</p> <p>Masukan Password</p> <p>Masuk</p> <p>Silakan masukan Email dan melanjutkan proses diagnosis jika memiliki akun silakan klik Di</p> <p><i>null</i></p>	 <p>Masuk</p> <p>Login Gagal! Email/Password Salah</p> <p>Masukan Email</p> <p>Masukan Password</p> <p>Masuk</p> <p>Silakan masukan Email dan melanjutkan proses diagnosis jika memiliki akun silakan klik Di</p> <p><i>error message</i></p>	Ditolak
<i>Email & Password</i>	 <p>Masuk</p> <p>inilisa@gmail.com</p> <p>Masukan Password</p> <p>Masuk</p> <p>Silakan masukan Email dan melanjutkan proses diagnosis jika memiliki akun silakan klik Di</p> <p><i>Data!=database</i></p>	 <p>Masuk</p> <p>Login Gagal! Email/Password Salah</p> <p>Masukan Email</p> <p>Masukan Password</p> <p>Masuk</p> <p>Silakan masukan Email dan melanjutkan proses diagnosis jika memiliki akun silakan klik Di</p> <p><i>Error message</i></p>	Ditolak
<i>Email & Password</i>	 <p>Masuk</p> <p>adi@gmail.com</p> <p>Masukan Password</p> <p>Masuk</p> <p>Silakan masukan Email dan melanjutkan proses diagnosis jika memiliki akun silakan klik Di</p> <p><i>Data=database</i></p>	 <p>Halaman Akun</p> <p>hai... Adi Isnanda Selamat Datang semoga anda sehat selalu</p> <p>adi@gmail.com</p> <p>Adi Isnanda</p> <p>089123456701</p> <p>Ubah</p> <p><i>Valid</i></p>	Diterima

h. Pengujian Halaman Daftar Akun (*user*)

Pengujian halaman daftar akun merupakan pengujian terhadap fungsionalitas halaman daftar akun. Tampilan hasil pengujian halaman daftar akun dapat dilihat pada Tabel 5.8 berikut.

Tabel 5.8 Pengujian halaman daftar akun (admin)

<i>Field</i>	<i>Input</i>	<i>Output</i>	<i>Kete- rangan</i>
<i>Email, nama lengkap, password, konfirmasi password.</i>	 <p>Daftar</p> <p>Masukan Email <small>Silakan isi form disamp untuk Login</small></p> <p>Masukan Nama Lengkap</p> <p>Password</p> <p>Konfirmasi Password</p> <p>Daftar</p> <p><i>null</i></p>	 <p>Daftar</p> <p>202101 <small>Silakan isi form disai untuk Login</small></p> <p>Masukan Nama Lengkap</p> <p>Nama Lengkap Harus Di Isi</p> <p>Konfirmasi Password</p> <p>Daftar</p> <p><i>error message</i></p>	Ditolak
<i>Email, nama lengkap, password, konfirmasi password.</i>	 <p>Daftar</p> <p>inirose@gmail.com <small>Silakan isi form untuk Login</small></p> <p>rose</p> <p>....</p> <p>....</p> <p>Daftar</p> <p><i>Data=database</i></p>	 <p>Masuk</p> <p>Akun berhasil tersimpan</p> <p>Masukan Email <small>Silakan m melanjutkan memiliki a</small></p> <p>Masukan Password</p> <p>Masuk</p> <p><i>Valid</i></p>	Diterima

5.1.3 Validasi Sistem

a. Validasi Sistem

Validasi sistem dilakukan untuk mencocokkan perancangan dengan sistem yang telah dibangun. Adapun hasil dari validasi sistem dapat dilihat pada Tabel 5.9 berikut.

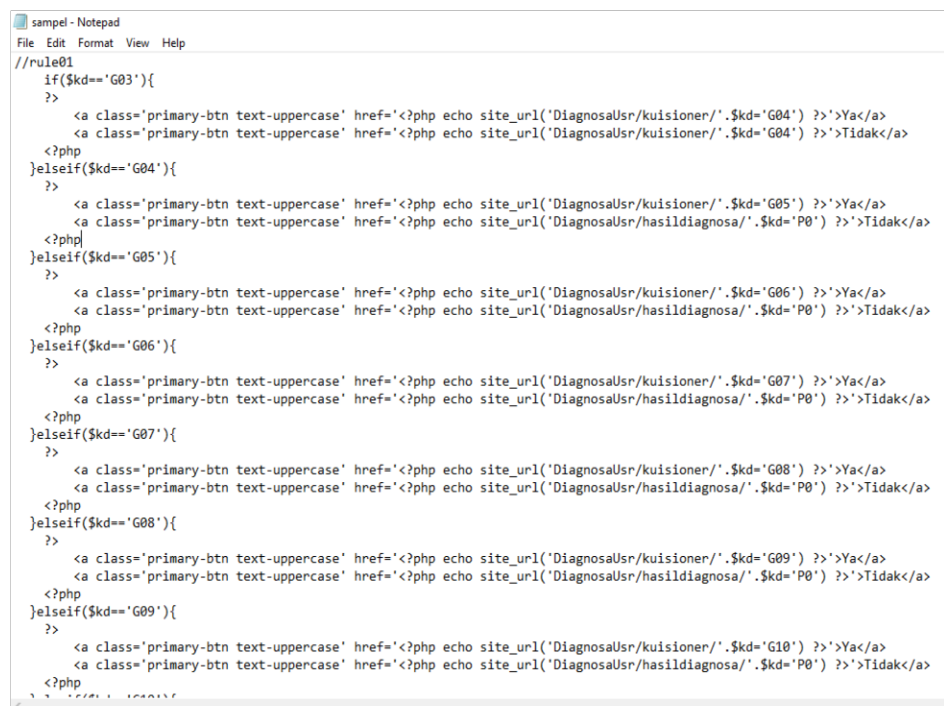
Tabel 5.9 Validasi sistem

No.	Pertanyaan	Peran-cangan	Sistem	Kete-rangan
1	Diare disertai lendir dan lebih dari 3 kali (G03) and Mual (G04) and Nyeri perut dibagian ulu hati disertai mual dan muntah (G05) and Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis (G06) and Hilang nafsu makan (G07) and Lemah (G08) and Lesu (G09) and Pegal pada otot (G10) and Flu disertai batuk pilek (G01) and Demam naik turun, suhu lebih dari 37.5 °C (G02) and Muntah (G12) and Air kencing gelap dikarenakan dehidrasi (G13)	Hepatitis A (P1)	Hepatitis A (P1)	<i>Valid</i>
2	Mual (G04) and Nyeri perut dibagian ulu hati disertai mual dan muntah (G05) and Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis (G06) and Hilang nafsu makan (G07) and Lesu (G09) and Pegal pada otot (G10) and Kembung (G11) and Muntah (G12) and Air kencing berwarna gelap karna dehidrasi (G13)	Hepatitis B (P2)	Hepatitis B (P2)	<i>Valid</i>

3	Nyeri perut dibagian ulu hati disertai mual dan muntah (G05) and Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis (G06) and Hilang nafsu makan (G07) and Lemah (G08) and Pegal pada otot (G10) and Kembung (G11) and (G12) and Air kencing gelap dikarenakan dehidrasi (G13)	Hepatitis C (P3)	Hepatitis C (P3)	<i>Valid</i>
---	---	------------------	------------------	--------------

Selanjutnya, sesuai dengan hasil tabel validasi diatas dapat ditentukan bagaimana sistem melakukan validasi *rules* saat *user* menjawab pertanyaan sesuai dengan gejala yang muncul. Dan akan dijelaskan sebagai berikut.

1. Penerapan *rules* 1, dapat dilihat pada Gambar 5.35 berikut



```
sampel - Notepad
File Edit Format View Help
//rule01
if($kd=='G03'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G04') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G04') ?>'>Tidak</a>
<?php
}elseif($kd=='G04'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G05') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
}elseif($kd=='G05'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G06') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
}elseif($kd=='G06'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G07') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
}elseif($kd=='G07'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G08') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
}elseif($kd=='G08'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G09') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
}elseif($kd=='G09'){
??
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/kuisisioner/'. $kd='G10') ?>'>Ya</a>
<a class='primary-btn text-uppercase' href='<?php echo site_url('DiagnosaUsr/hasilidiagnosa/'. $kd='P0') ?>'>Tidak</a>
<?php
```

Gambar 5.35 Penerapan *rules* 1

5.2 Pembahasan

5.2.1 Penerapan Perhitungan Algoritma *Certainty Factor*.

Pada penerapan algoritma *certainty factor* digunakan rumus perhitungan:

$$CF_{\text{kombinasi}} = CF[H,e]_1 + CF[H,e]_2 * (1 - CF[H,e]_1)$$

Keterangan:

CF[H,e] : faktor kepastian.

CF *input* pakar : nilai kepercayaan yang diperoleh dari nilai data gejala dari pakar/dokter.

CF *input user* : nilai kepercayaan yang diperoleh dari pemilihan data diagnosis berupa iya atau tidak yang diisi oleh pengguna.

CF_{kombinasi} : diperoleh dengan mengkombinasikan CF *input* pakar dan CF *input* user yang merupakan nilai yang didapat dari kepercayaan setiap *rules*.

1. Penerapan Algoritma pada perhitungan Hepatitis A

a. Penerapan perhitungan Hepatitis A menggunakan sampel akun ahmad@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel CF *input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.10 CF input pakar hepatitis A

Kode Gejala	Nama gejala	Input pakar (%)
G03	Diare disertai lendir dan lebih dari 3 kali	0,24
G04	Mual	0,21
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,21

G06	Mata kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G07	Hilang nafsu makan	0,14
G08	Lemah	0,14
G09	Lesu	0,14
G10	Pegal pada otot	0,14
G01	Flu disertai batuk pilek	0,12
G02	Demam naik turun, suhu lebih dari 37.5 °C	0,12
G12	Muntah	0,12
G13	Kulit kuning sebagai tanda paling spesifik pada penyakit hepatitis	0,11

2. Kemudian, menentukan nilai bobot *CF input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.11 *CF input user hepatitis A*

Input user	Keterangan
0	Tidak
1	Iya

3. Langkah berikutnya adalah mencari *CF kombinasi* yang diperoleh dengan mengalikan *CF user* dan *CF pakar*. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.12 *CF kombinasi Hepatitis A*

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Diare disertai lendir dan lebih dari 3 kali	1	0,24	0,24
2	Mual	1	0,21	0,21
3	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,21	0,21

4	Mata kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
5	Hilang nafsu makan	1	0,14	0,14
6	Lemah	1	0,14	0,14
7	Lesu	1	0,14	0,14
8	Pegal pada otot	1	0,14	0,14
9	Flu disertai batuk pilek	1	0,12	0,12
10	Demam naik turun, suhu lebih dari 37.5 °C	1	0,12	0,12
11	Muntah	1	0,12	0,12
12	Air kencing gelap dikarenakan dehidrasi	1	0,11	0,11

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan:

$$CF_{\text{kombinasi}} = CF[H,e]1 + CF[H,e]2 * (1 - CF[H,e]1)$$

1. $CF_{k1} = CF[H,e]1 + CF[H,e]2 * (1 - CF[H,e]1)$
 $= 0,21 + 0,21 * (1-0,12)$
 $= 0,3948$
2. $CF_{k2} = CF_{k1} + CF[H,e]3 * (1 - CF_{k1})$
 $= 0,3948 + 0,21 * (1-0,3948)$
 $= 0,521892$
3. $CF_{k3} = CF_{k2} + CF[H,e]4 * (1 - CF_{k2})$
 $= 0,521892 + 0,25 * (1-0,521892)$

$$= 0,57313972$$

4. $CFk4 = CFk3 + CF[H,e]5 * (1 - CFk3)$
 $= 0,57313972 + 0,14 * (1 - 0,57313972)$
 $= 0,70829566$
5. $CFk5 = CFk4 + CF[H,e]6 * (1 - CFk4)$
 $= 0,70829566 + 0,14 * (1 - 0,70829566)$
 $= 0,7491342676$
6. $CFk6 = CFk5 + CF[H,e]7 * (1 - CFk5)$
 $= 0,7491342676 + 0,14 * (1 - 0,7491342676)$
 $= 0,784255470136$
7. $CFk7 = CFk6 + CF[H,e]8 * (1 - CFk6)$
 $= 0,784255470136 + 0,14 * (1 - 0,784255470136)$
 $= 0,81445970431696$
8. $CFk8 = CFk7 + CF[H,e]9 * (1 - CFk7)$
 $= 0,81445970431696 + 0,12 * (1 - 0,81445970431696)$
 $= 0,8367245397989248$
9. $CFk9 = CFk8 + CF[H,e]10 * (1 - CFk8)$
 $= 0,8367245397989248 + 0,12 * (1 - 0,8367245397989248)$
 $= 0,856317595023053824$
10. $CFk10 = CFk9 + CF[H,e]11 * (1 - CFk10)$
 $= 0,856317595023053824 + 0,12 * (1 -$
 $0,856317595023053824)$
 $= 0,87355948362028736512$
11. $CFk11 = CFk10 + CF[H,e]12 * (1 - CFk11)$
 $= 0,87355948362028736512 + 0,11 * (1 -$
 $0,87355948362028736512)$
 $= 0,8874679404220557549568$

5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $CF_{kombinasi} * 100\%$. Sehingga dihasilkan perhitungan sebesar **88,7%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis A.

b. Penerapan perhitungan Hepatitis A menggunakan sampel akun marchel@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel CF *input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.13 CF input pakar hepatitis A

Kode Gejala	Nama gejala	Input pakar (%)
G03	Diare disertai lendir dan lebih dari 3 kali	0,24
G04	Mual	0,21
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,21
G06	Mata kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G07	Hilang nafsu makan	0,14
G08	Lemah	0,14
G09	Lesu	0,14
G10	Pegal pada otot	0,14
G01	Flu disertai batuk pilek	0,12
G02	Demam naik turun, suhu lebih dari 37.5 °C	0,12
G12	Muntah	0,12
G13	Kulit kuning sebagai tanda paling spesifik pada penyakit hepatitis	0,11

2. Kemudian, menentukan nilai bobot CF *input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.14 CF input user hepatitis A

Input user	Keterangan
0	Tidak

1	Iya
---	-----

3. Langkah berikutnya adalah mencari CFkombinasi yang diperoleh dengan mengalikan CF_{user} dan CF_{pakar} . Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.15 CF kombinasi Hepatitis A

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Diare disertai lendir dan lebih dari 3 kali	1	0,24	0,24
2	Mual	1	0,21	0,21
3	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,21	0,21
4	Mata kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
5	Hilang nafsu makan	1	0,14	0,14
6	Lemah	1	0,14	0,14
7	Lesu	1	0,14	0,14
8	Pegal pada oto	1	0,14	0,14
9	Flu disertai batuk pilek	1	0,12	0,12
10	Demam naik turun, suhu lebih dari 37.5 °C	1	0,12	0,12
11	Muntah	1	0,12	0,12
12	Air kencing gelap dikarenakan dehidrasi	1	0,11	0,11

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan:

$$\text{CF}_{\text{kombinasi}} = \text{CF}[\text{H,e}]1 + \text{CF}[\text{H,e}]2 * (1 - \text{CF}[\text{H,e}]1)$$

1. $\text{CF}_{k1} = \text{CF}[\text{H,e}]1 + \text{CF}[\text{H,e}]2 * (1 - \text{CF}[\text{H,e}]1)$
 $= 0,21 + 0,21 * (1-0,12)$
 $= 0,3948$
2. $\text{CF}_{k2} = \text{CF}_{k1} + \text{CF}[\text{H,e}]3 * (1 - \text{CF}_{k1})$
 $= 0,3948 + 0,21 * (1-0,3948)$
 $= 0,521892$
3. $\text{CF}_{k3} = \text{CF}_{k2} + \text{CF}[\text{H,e}]4 * (1 - \text{CF}_{k2})$
 $= 0,521892 + 0,25 * (1-0,521892)$
 $= 0,57313972$
4. $\text{CF}_{k4} = \text{CF}_{k3} + \text{CF}[\text{H,e}]5 * (1 - \text{CF}_{k3})$
 $= 0,57313972 + 0,14 * (1-0,57313972)$
 $= 0,70829566$
5. $\text{CF}_{k5} = \text{CF}_{k4} + \text{CF}[\text{H,e}]6 * (1 - \text{CF}_{k4})$
 $= 0,70829566 + 0,14 * (1-0,70829566)$
 $= 0,7491342676$
6. $\text{CF}_{k6} = \text{CF}_{k5} + \text{CF}[\text{H,e}]7 * (1 - \text{CF}_{k5})$
 $= 0,7491342676 + 0,14 * (1-0,7491342676)$
 $= 0,784255470136$
7. $\text{CF}_{k7} = \text{CF}_{k6} + \text{CF}[\text{H,e}]8 * (1 - \text{CF}_{k6})$
 $= 0,784255470136 + 0,14 * (1-0,784255470136)$
 $= 0,81445970431696$
8. $\text{CF}_{k8} = \text{CF}_{k7} + \text{CF}[\text{H,e}]9 * (1 - \text{CF}_{k7})$
 $= 0,81445970431696 + 0,12 * (1-0,81445970431696)$
 $= 0,8367245397989248$
9. $\text{CF}_{k9} = \text{CF}_{k8} + \text{CF}[\text{H,e}]10 * (1 - \text{CF}_{k8})$
 $= 0,8367245397989248 + 0,12 * (1-0,8367245397989248)$

$$= 0,856317595023053824$$

$$10. CFk10 = CFk9 + CF[H,e]11 * (1 - CFk10)$$

$$= 0,856317595023053824 + 0,12 * (1 - 0,856317595023053824)$$

$$= 0,87355948362028736512$$

$$11. CFk11 = CFk10 + CF[H,e]12 * (1 - CFk11)$$

$$= 0,87355948362028736512 + 0,11 * (1 - 0,87355948362028736512)$$

$$= 0,8874679404220557549568$$

5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $CF_{kombinasi} * 100\%$. Sehingga dihasilkan perhitungan sebesar **88,7%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis A.

Dari 2 sampel perhitungan diatas dengan menggunakan 2 akun yang berbeda, maka diperoleh nilai keyakinan tetap pengguna mengidap penyakit Hepatitis A sebesar **88,7%**.

2. Penerapan Algoritma pada perhitungan Hepatitis B.

a. Penerapan perhitungan Hepatitis B menggunakan sampel akun akhmad@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel CF *input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.16 CF input pakar Hepatitis B

Kode Gejala	Nama gejala	Input pakar (%)
G04	Mual	0,22
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,22

G06	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G07	Hilang nafsu makan	0,15
G09	Lesu	0,15
G10	Pegal pada otot	0,15
G11	Kembung	0,12
G12	Muntah	0,13
G13	Air kencing gelap dikarenakan dehidrasi	0,12

2. Kemudian, menentukan nilai bobot CF *input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.17 CF input user Hepatitis B

Input user	Keterangan
0	Tidak
1	Iya

3. Langkah berikutnya adalah mencari CF kombinasi yang diperoleh dengan mengalikan CF *user* dan CF pakar. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.18 CF kombinasi Hepatitis B

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Mual	1	0,22	0,22
2	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,22	0,22
3	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
4	Hilang nafsu makan	1	0,15	0,15

5	Lesu	1	0,15	0,15
6	Pegal pada otot	1	0,15	0,15
7	Kembung	1	0,12	0,12
8	Muntah	1	0,13	0,13
9	Air kencing gelap dikarenakan dehidrasi	1	0,12	0,12

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan.

$$CF_{\text{kombinasi}} = CF[H,e]_1 + CF[H,e]_2 * (1 - CF[H,e]_1)$$

$$1. CF_{k1} = CF[H,e]_1 + CF[H,e]_2 * (1 - CF[H,e]_1)$$

$$= 0,22 + 0,22 * (1-0,22)$$

$$= 0,3916$$

$$2. CF_{k2} = CF_{k1} + CF[H,e]_3 * (1 - CF_{k1})$$

$$= 0,3916 + 0,25 * (1-0,3916)$$

$$= 0,5437$$

$$3. CF_{k3} = CF_{k2} + CF[H,e]_4 * (1 - CF_{k2})$$

$$= 0,5437 + 0,15 * (1-0,5437)$$

$$= 0,612145$$

$$4. CF_{k4} = CF_{k3} + CF[H,e]_5 * (1 - CF_{k3})$$

$$= 0,612145 + 0,15 * (1-0,612145)$$

$$= 0,67032325$$

$$5. CF_{k5} = CF_{k4} + CF[H,e]_6 * (1 - CF_{k4})$$

$$= 0,67032325 + 0,15 * (1-0,67032325)$$

$$= 0,7197747625$$

$$6. CF_{k6} = CF_{k5} + CF[H,e]_7 * (1 - CF_{k5})$$

$$= 0,7197747625 + 0,12 * (1-0,7197747625)$$

$$= 0,753401791$$

$$7. CF_{k7} = CF_{k6} + CF[H,e]_8 * (1 - CF_{k6})$$

$$= 0,753401791 + 0,13 * (1-0,753401791)$$

$$=0,78545955817$$

$$8. CFk8 = CFk7 + CF[H,e]9 * (1 - CFk7)$$

$$= 0,78545955817 + 0,12 * (1-0,78545955817)$$

$$= 0,8112044111896$$

5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $CF_{kombinasi} * 100\%$. Sehingga dihasilkan perhitungan sebesar **81,1%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis B.

b. Penerapan perhitungan Hepatitis B menggunakan sampel akun marchel@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel *CF input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.19 CF input pakar Hepatitis B

Kode Gejala	Nama gejala	Input pakar (%)
G04	Mual	0,22
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,22
G06	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G07	Hilang nafsu makan	0,15
G09	Lesu	0,15
G10	Pegal pada otot	0,15
G11	Kembung	0,12
G12	Muntah	0,13
G13	Air kencing gelap dikarenakan dehidrasi	0,12

2. Kemudian, menentukan nilai bobot CF *input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.20 CF input user Hepatitis B

Input user	Keterangan
0	Tidak
1	Iya

3. Langkah berikutnya adalah mencari CF kombinasi yang diperoleh dengan mengalikan CF *user* dan CF pakar. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.21 CF kombinasi Hepatitis B

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Mual	1	0,22	0,22
2	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,22	0,22
3	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
4	Hilang nafsu makan	1	0,15	0,15
5	Lesu	1	0,15	0,15
6	Pegal pada otot	1	0,15	0,15
7	Kembung	1	0,12	0,12
8	Muntah	1	0,13	0,13
9	Air kencing gelap dikarenakan dehidrasi	1	0,12	0,12

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan.

$$\text{CF}_{\text{kombinasi}} = \text{CF}[\text{H,e}]1 + \text{CF}[\text{H,e}]2 * (1 - \text{CF}[\text{H,e}]1)$$

1. $\text{CF}_{k1} = \text{CF}[\text{H,e}]1 + \text{CF}[\text{H,e}]2 * (1 - \text{CF}[\text{H,e}]1)$
 $= 0,22 + 0,22 * (1-0,22)$
 $= 0,3916$
 2. $\text{CF}_{k2} = \text{CF}_{k1} + \text{CF}[\text{H,e}]3 * (1 - \text{CF}_{k1})$
 $= 0,3916 + 0,25 * (1-0,3916)$
 $= 0,5437$
 3. $\text{CF}_{k3} = \text{CF}_{k2} + \text{CF}[\text{H,e}]4 * (1 - \text{CF}_{k2})$
 $= 0,5437 + 0,15 * (1-0,5437)$
 $= 0,612145$
 4. $\text{CF}_{k4} = \text{CF}_{k3} + \text{CF}[\text{H,e}]5 * (1 - \text{CF}_{k3})$
 $= 0,612145 + 0,15 * (1-0,612145)$
 $= 0,67032325$
 5. $\text{CF}_{k5} = \text{CF}_{k4} + \text{CF}[\text{H,e}]6 * (1 - \text{CF}_{k4})$
 $= 0,67032325 + 0,15 * (1-0,67032325)$
 $= 0,7197747625$
 6. $\text{CF}_{k6} = \text{CF}_{k5} + \text{CF}[\text{H,e}]7 * (1 - \text{CF}_{k5})$
 $= 0,7197747625 + 0,12 * (1-0,7197747625)$
 $= 0,753401791$
 7. $\text{CF}_{k7} = \text{CF}_{k6} + \text{CF}[\text{H,e}]8 * (1 - \text{CF}_{k6})$
 $= 0,753401791 + 0,13 * (1-0,753401791)$
 $= 0,78545955817$
 8. $\text{CF}_{k8} = \text{CF}_{k7} + \text{CF}[\text{H,e}]9 * (1 - \text{CF}_{k7})$
 $= 0,78545955817 + 0,12 * (1-0,78545955817)$
 $= 0,8112044111896$
5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $\text{CF}_{\text{kombinasi}} * 100\%$.

Sehingga dihasilkan perhitungan sebesar **81,1%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis B.

Dari 2 sampel perhitungan diatas dengan menggunakan 2 akun yang berbeda, maka diperoleh nilai keyakinan tetap pengguna mengidap penyakit Hepatitis B sebesar **81,1%**.

3. Penerapan Algoritma pada perhitungan Hepatitis C.

a. Penerapan perhitungan Hepatitis C menggunakan sampel akun akhmad@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel CF *input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.22 CF input pakar Hepatitis C

Kode Gejala	Nama gejala	Input pakar (%)
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,22
G06	Mual	0,22
G07	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G08	Lemah	0,16
G10	Pegal pada otot	0,16
G11	Kembung	0,16
G12	Muntah	0,12
G13	Air kencing gelap dikarenakan dehidrasi	0,14

2. Kemudian, menentukan nilai bobot CF *input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.23 CF input user Hepatitis C

Input user	Keterangan
0	Tidak
1	Iya

3. Langkah berikutnya adalah mencari CF kombinasi yang diperoleh dengan mengalikan CF *user* dan CF pakar. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.24 CF kombinasi Hepatitis C

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,22	0,22
2	Mual	1	0,22	0,22
3	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
4	Lemah	1	0,16	0,16
5	Pegal pada otot	1	0,16	0,16
6	Kembung	1	0,16	0,16
7	Muntah	1	0,12	0,12
8	Air kencing gelap dikarenakan dehidrasi	1	0,14	0,14

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan:

$$CF_{\text{kombinasi}} = CF[H,e]_1 + CF[H,e]_2 * (1 - CF[H,e]_1)$$

$$1. CFk1 = CF[H,e]1 + CF[H,e]2 * (1 - CF[H,e]1)$$

$$= 0,22 + 0,22 * (1-0,22)$$

$$= 0,3916$$

$$2. CFk2 = CFk1 + CF[H,e]3 * (1 - CFk1)$$

$$= 0,3916 + 0,25 * (1-0,3916)$$

$$= 0,5437$$

$$3. CFk3 = CFk2 + CF[H,e]4 * (1 - CFk2)$$

$$= 0,5437 + 0,16 * (1-0,5437)$$

$$= 0,616708$$

$$4. CFk4 = CFk3 + CF[H,e]5 * (1 - CFk3)$$

$$= 0,616708 + 0,16 * (1-0,616708)$$

$$= 0,67803472$$

$$5. CFk5 = CFk4 + CF[H,e]6 * (1 - CFk4)$$

$$= 0,67803472 + 0,16 * (1-0,67803472)$$

$$= 0,7295491648$$

$$6. CFk6 = CFk5 + CF[H,e]7 * (1 - CFk5)$$

$$= 0,7295491648 + 0,12 * (1-0,7295491648)$$

$$= 0,762003265024$$

$$7. CFk7 = CFk6 + CF[H,e]8 * (1 - CFk6)$$

$$= 0,762003265024 + 0,14 * (1-0,762003265024)$$

$$= 0,79532280792064$$

5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $CF_{kombinasi} * 100\%$. Sehingga dihasilkan perhitungan sebesar **79,5%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis C.

- b. Penerapan perhitungan Hepatitis C menggunakan sampel akun marchel@gmail.com dapat dilihat pada perhitungan berikut:

1. Sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel CF *input* pakar. Yang berisi

nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.

Tabel 5.25 CF input pakar Hepatitis C

Kode Gejala	Nama gejala	Input pakar (%)
G05	Nyeri perut dibagian ulu hati disertai mual dan muntah	0,22
G06	Mual	0,22
G07	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	0,25
G08	Lemah	0,16
G10	Pegal pada otot	0,16
G11	Kembung	0,16
G12	Muntah	0,12
G13	Air kencing gelap dikarenakan dehidrasi	0,14

2. Kemudian, menentukan nilai bobot CF *input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

Tabel 5.26 CF input user Hepatitis C

Input user	Keterangan
0	Tidak
1	Iya

3. Langkah berikutnya adalah mencari CF kombinasi yang diperoleh dengan mengalikan CF *user* dan CF pakar. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.

Tabel 5.27 CF kombinasi Hepatitis C

No	Nama gejala	Cf user	Cf pakar	Cf kombinasi
1	Nyeri perut dibagian ulu hati disertai mual dan muntah	1	0,22	0,22
2	Mual	1	0,22	0,22
3	Kulit kuning sebagai tanda paling efektif pada penyakit hepatitis	1	0,25	0,25
4	Lemah	1	0,16	0,16
5	Pegal pada otot	1	0,16	0,16
6	Kembung	1	0,16	0,16
7	Muntah	1	0,12	0,12
8	Air kencing gelap dikarenakan dehidrasi	1	0,14	0,14

4. Selanjutnya, berdasarkan nilai CF kombinasi pada tabel diatas, ditentukan nilai CF gabungan dari setiap *rules* dengan menggunakan aturan:

$$CF_{\text{kombinasi}} = CF[H,e]1 + CF[H,e]2 * (1 - CF[H,e]1)$$

1. $CF_{k1} = CF[H,e]1 + CF[H,e]2 * (1 - CF[H,e]1)$
 $= 0,22 + 0,22 * (1-0,22)$
 $= 0,3916$
2. $CF_{k2} = CF_{k1} + CF[H,e]3 * (1 - CF_{k1})$
 $= 0,3916 + 0,25 * (1-0,3916)$
 $= 0,5437$
3. $CF_{k3} = CF_{k2} + CF[H,e]4 * (1 - CF_{k2})$
 $= 0,5437 + 0,16 * (1-0,5437)$
 $= 0,616708$

$$\begin{aligned}
4. \quad CFk4 &= CFk3 + CF[H,e]5 * (1 - CFk3) \\
&= 0,616708 + 0,16 * (1-0,616708) \\
&= 0,67803472 \\
5. \quad CFk5 &= CFk4 + CF[H,e]6 * (1 - CFk4) \\
&= 0,67803472 + 0,16 * (1-0,67803472) \\
&= 0,7295491648 \\
6. \quad CFk6 &= CFk5 + CF[H,e]7 * (1 - CFk5) \\
&= 0,7295491648 + 0,12 * (1-0,7295491648) \\
&= 0,762003265024 \\
7. \quad CFk7 &= CFk6 + CF[H,e]8 * (1 - CFk6) \\
&= 0,762003265024 + 0,14 * (1-0,762003265024) \\
&= 0,79532280792064
\end{aligned}$$

5. Selanjutnya, untuk mendapatkan nilai presentase identifikasi penyakit, dilakukan perhitungan $CF_{\text{kombinasi}} * 100\%$. Sehingga dihasilkan perhitungan sebesar **79,5%** sebagai tingkat keyakinan pengguna mengidap penyakit Hepatitis jenis C.

Dari 2 sampel perhitungan diatas dengan menggunakan 2 akun yang berbeda, maka diperoleh nilai keyakinan tetap pengguna mengidap penyakit Hepatitis C sebesar **79,5%**.

4. Penerapan Algoritma Apabila Prediksi Diluar Sistem.

1. Seperti yang sudah dijelaskan sebelumnya, dimana dalam perhitungan sesuai dengan diagnosis yang telah diisi oleh pengguna pada proses diagnosis, diperoleh tabel *CF input* pakar. Yang berisi nama gejala serta bobot gejala yang didasarkan oleh penilaian pakar.
2. Kemudian, menentukan nilai bobot *CF input user*. Tabel ini berisi bobot dari pilihan diagnosis berupa iya dan tidak yang memiliki bobot tersendiri.

3. Langkah berikutnya adalah mencari CF kombinasi yang diperoleh dengan mengalikan CF *user* dan CF pakar. Sedangkan gejala yang tidak terpilih sistem akan otomatis memberikan nilai 0.
4. Apabila pengguna sudah memasukan semua gejala yang dirasakan, tapi dari gejala-gejala tersebut tidak sesuai dengan *record database* sistem pakar maka nilai dari CF kombinasi, CF *user*, dan CF pakar akan secara otomatis berisi 0. Sehingga kondisi tersebut tidak memenuhi *rules* yang ada sehingga diperoleh kondisi *output* “Diluar Prediksi Sistem”.

BAB VI

PENUTUP

6.1 Kesimpulan

Dalam penelitian yang dilakukan menghasilkan beberapa kesimpulan. Adapun kesimpulan yang didapatkan adalah sebagai berikut:

1. Penerapan Sistem Pakar Diagnosis Awal Penyakit Hepatitis dengan Metode *Forward Chaining* dan *Certainty Factor* telah dapat dibangun dan memberikan informasi kepada user berdasarkan gejala-gejala fisik yang diderita oleh pasien, dengan menggunakan metode *Forward Chaining* dan *Certainty Factor*.
2. Pada Sistem Pakar Diagnosis Awal Penyakit Hepatitis dengan Metode *Forward Chaining* dan *Certainty Factor* ini terdapat halaman *Admin* untuk mengelola sumber daya sistem seperti data diagnosis, data gejala, data berita, dan data *user*.
3. Berdasarkan perbandingan hasil diagnosis penyakit pada sistem dengan pakar maka tingkat keakuratan sistem pada jenis Hepatitis A sebesar 88,7%, Hepatitis B sebesar 81,8%, dan Hepatitis C sebesar 79,5%.

6.2 Saran

Berdasarkan pembuatan Sistem Pakar Diagnosis Awal Penyakit Hepatitis Metode *Forward Chaining* dan *Certainty*, maka untuk penelitian selanjutnya disarankan agar sistem ini dapat dikembangkan menjadi lebih baik, yaitu:

1. Menambahkan suatu metode untuk penentuan komposisi obat sesuai dengan gejala dan penyakit.
2. Menambahkan data penyakit yang tidak hanya berasal dari gejala fisik pasien.
3. Menambahkan fitur yang dapat menghubungkan *user* dengan instalasi kesehatan terdekat.

DAFTAR PUSTAKA

- [1] Badan Penelitian dan Pengembangan Kesehatan, Kementerian Kesehatan RI, “Riset Data Kesehatan Tahun 2018” Tim Labmandat Balitbangkes, 2018.
- [2] WHO Indonesia, World Health Organization, “Hari Hepatitis Sedunia”, 2020
- [3] A. Ramdhani, R.R. Isnanto, I.P. Windasari, “Pengembangan Sistem Pakar Untuk Diagnosis Penyakit Hepatitis Berbasis Web Menggunakan Metode Certainty Factor”, Universitas Diponegoro, vol.9, no.3, pp.58-64, Januari 2016.
- [4] Pusat Data Dan Informasi Kementerian Kesehatan RI, Kementerian Kesehatan RI, “Situasi Dan Analisis Hepatitis, Pekan Peduli Hepatitis” Pusat Data Dan Informasi, 2016.
- [5] L.N. Widi, “Sistem Pakar Diagnosis Awal Penyakit Hepatitis Berbasis Web Dengan Metode Certainty Factor (CF)”, STMIK Sinar Nusantara Surakarta, November 2016.
- [6] D.T Yuwono, A. Fadlil, Sumardi, “Penerapan Metode Forward Chaining Dan Certainty Factor Pada Sistem Pakar Diagnosishama Anggrek Coelogyne Pandurata”, Universitas Ahmad Dahlan Yogyakarta, vol.4, no.2, September 2017.
- [7] H. Susilo, “Sistem Pakar Metode Forward Chaining Dan Certainty Factor Untuk Mengidentifikasi Penyakit Pertusis Pada Anak”, STIKES Syedza Saintika, vol.1, no.2, Juni 2018.
- [8] Sari, “Implementasi Sistem Pakar Untuk Diagnosis Awal Penyakit Hepatitis A,B,C Menggunakan Tools Expert System Builder”, Universitas Nasional, vol.8,no.2, pp.173-184, November 2016.
- [9] E.M. Nugmadi, I. Ratna, “Sistem Pakar Mendiagnosis Penyakit Hepatitis Menggunakan Metode Certainty Factor Berbasis Web”, Universitas Muhammadiyah Sidoarjo, 2016.

- [10] A.T. Novarina, E. Santoso, Indriati, “Sistem Pakar Diagnosis Penyakit Hepatitis Menggunakan Metode Dempster Shafer”, Universitas Brawijaya, vol.2, no.6, pp 2252-2258, Juni 2018.
- [11] R. Rosnelly, “Sistem Pakar Dan Teori”, Yogyakarta: Andi Offset, 2014.
- [12] Pusat Data Dan Informasi Kementerian Kesehatan RI, Kementerian Kesehatan RI, “Situasi Penyakit Hepatitis B Di Indonesia 2017” Pusat Data Dan Informasi, 2017.
- [13] Direktorat Jendral PP & PL Kementerian Kesehatan RI, “Pedoman Pengendalian Hepatitis Virus”, Kementerian Kesehatan RI, 2015.
- [14] Gilmore, W. Jason 2008. Beginning PHP and MySQL From Novice to Professional. Apress.
- [15] Komputer, W. 2010. Panduan Belajar MySQL Database Server. Mediakita.
- [16] B. Raharjo, I. Heryanto dan E. RK, Modul Pemrograman WEB, Bandung: Modula, 2010.
- [17] A. Nugroho, Rekayasa Perangkat Lunak dengan UML dan Java, Yogyakarta: Andi, 2005.
- [18] Prabowo Pudjo Widodo, Herlawati. Menggunakan UML. Bandung: Informatika, 2011.
- [19] Fowler, Martin. UML Distilled: Panduan Singkat Bahasa Pemodelan Objek Standar. Yogyakarta: Andi Publisher, 2005.
- [20] Sugiarti, Yuni. Analisis dan Perancangan UML (Unified Modeling Language). Yogyakarta: Graha Ilmu. 2013.

DAFTAR LAMPIRAN

Lampiran 1 Surat Kesepakatan Bimbingan Tugas Akhir Pembimbing 1

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Mukhammad Sabda Abdulloh
NIM : 16090106
Program studi : DIV Teknik Informatika


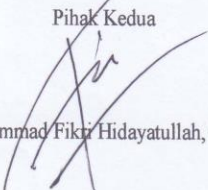
Pihak Kedua

Nama : Muhammad Fikri Hidayatullah, S.T., M.Kom
Status : Dosen
NIPY : 09.016.307
Jabatan Fungsional : Asisten Ahli

Pada hari ini senin tanggal 24 Februari 2020 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama Wajib melakukan bimbingan Tugas Akhir sekurang-kurangnya 1 (satu) kali dalam 1 (satu) minggu, adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Tegal, 24 Februari 2020

<p>Pihak Pertama</p>  Mukhammad Sabda Abdulloh	<p>Pihak Kedua</p>  Muhammad Fikri Hidayatullah, S.T., M.Kom
--	---

Lampiran 2 Surat Kesepakatan Bimbingan Tugas Akhir Pembimbing 2

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Mukhammad Sabda Abdulloh

NIM : 16090106

Program studi : DIV Teknik Informatika

Pihak Kedua

Nama : Sharfina Febbi Handayani, M.Kom

Status : Dosen

NIPY : 08.020.451

Jabatan Fungsional : Tenaga Pengajar

Pada hari ini kamis tanggal 22 April 2021 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama Wajib melakukan bimbingan Tugas Akhir sekurang-kurangnya 1 (satu) kali dalam 1 (satu) minggu, adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

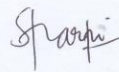
Tegal, 22 April 2021

Pihak Pertama



Mukhammad Sabda Abdulloh

Pihak Kedua



Sharfina Febbi Handayani, M.Kom

Lampiran 3 Lembar Bimbingan Tugas Akhir Pembimbing 1




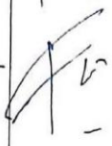



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mukhammad Sabda Abdulloh
 NIM : 16090106
 No. Ponsel : 0878 9988 9628
 Judul TA : Penerapan Sistem Pakar Untuk Diagnosis Penyakit
 Hepatitis Menggunakan Metode *Forward Chaining* Dan
Certainty Factor

Dosen Pembimbing I: Muhammad Fikri Hidayatullah, S.T., M.Kom

No.	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	Kanid	Konsep TA	- Cari lagi pakar untuk penyakit Hepatitis. - Penemuan kasus Konsep TA - konsep TA sudah ok	
2.	25 Juli 2020	Pemeriksaan metode-metode.	- pelajaran ttg metode-metode yang digunakan.	
3.	30 Juli		- perhitungan certainty factor	

3.	30 Juni 2020	Perhitungan Certainty factor		
4.	6/3 2021	Penulisan Tinjauan Pustaka	Buat kembali Certif.	
5.	24/3	Bab II-III	- Di buat lebih sistematis lagi penulisan di Tinjauan pustaka - teori Cleaning form & PC - UML	
6.	25/3	Bab III-IV	- Realisasi di program Pemesan Libiwa Certif	
7.	7/4	Bab IV	UML paradi	

8.	17/6	- Lanjutkan - Aplikasi selesaikan.		Fikri
9.	7 Juli 2021	Progres laporan.	- Gunakan kata baku = diagnosis, analisis. - Lengkapi daftar pustaka, lampiran	Fikri
10.	15 Juli 2021	Progres laporan	- Perbaiki penulisan naskah. - Lengkapi lampiran.	Fikri
11.	17 Juli 2021	Acc, siap maju sidang.		Fikri

Tegal,.....2021

Dosen Pembimbing I

Fikri

Muhammad Fikri Hidayatullah, S.T., M.Kom

NIPY. 09.016.307

Lampiran 4 Lembar Bimbingan Tugas Akhir Pembimbing 2



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mukhammad Sabda Abdulloh
 NIM : 16090106
 No. Ponsel : 082241058926
 Judul TA : Penerapan Sistem Pakar Untuk Diagnosa Awal Penyakit
 Hepatitis Menggunakan Metode *Forward Chaining* dan
Certainty Factor

Dosen Pembimbing II : Sharfina Febbi Handayani M.Kom

No.	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	27 April 2021	Bab I Pendahuluan Bab II Tinjauan Pustaka Bab III Landasan Teori Bab IV	- Istilah asing dan pemilihan kata yg sesuai - Redaksi penulisan awal kalimat - rujukan penentuan bobot pakar perlu ditambahkan mengacu ke literatur mang. - penggunaan kata 'penggun' disesuaikan dg gambar dan siapa aktornya	SA
2.	9 Juni 2021	Bab IV Desain & Perancangan	- Usecase ditambahkan untuk aktor user (hasil diagnosa) - tipe data dipelajari lagi mengapa memilih tipe data tsb	SA

3.	11 Juni 2021	Demo Aplikasi	<ul style="list-style-type: none"> - halaman tambah user di Admin masih belum ada - Rekam diagnosa 4 user dibuat report - Modul diagnosa belum - Ubah gambar user belum 	Sf.
4.	18 Juni 2021	Demo Aplikasi	<ul style="list-style-type: none"> - Informasi penggunaan aplikasi pd profil user - Ubah profil user belum berjalan. - Kontak ditambahkan pd halaman user - Kuisioner dilengkapi sampai 615 dan rules pohon keputusan di lanjut diselesaikan sampai 615. 	Sf.
5.	6 Juli 2021	- Demo aplikasi - Bab 5 Hasil dan Pembahasan - Bab 6 Penutup.	<ul style="list-style-type: none"> - Aplikasi sudah ok - Ditambahkan frase yg menjelaskan sub poin informasi. - Ditambahkan ss rules pada saat sist memvalidasi saat user melakukan pendaftaran - perbaikan kalimat dan pengalihan kata yg lebih baik. 	Sf.
6.	7 Juli 2021	- Bab 5 - Bab 6	<ul style="list-style-type: none"> - semua laporan dari bab 1-6 diuploadkan saku. 	Sf.

7.	11 Juli 2021	- progres laporan tugas akhir	- gambar usecase diperbaiki, dan halaman include disiapkan usecase ke usecase login. - diberi prototipe narasi sesuai gambar yang disulitkan. - dikembangkan desain mockup sesuai halaman.	SF
8.	13 Juli 2021	- progres laporan bab 1-6.	- perbaikan hasil kesimpulan ditulis sesuai hasil observasi wawancara.	SF
9.	16 Juli 2021	- Acc.		SF

Tegal,16 Juli.....2021

Dosen Pembimbing II

Sharfina

Sharfina Febbi Handayani M.Kom

NIPY. 08.020.451

Lampiran 5 Surat Permohonan Ijin Penelitian



Yayasan Pendidikan Harapan Bersama

PoliTekniK Harapan Bersama

PROGRAM STUDI D IV TEKNIK INFORMATIKA

Kampus I : Jl. Mataram No. 9 Tegal 52142 Telp. 0283-352000 Fax. 0283-353353

Website : www.poltektegal.ac.id Email : d4.informatika@poltektegal.ac.id

Nomor : 171.03/TL.PHB/IV/2021
Lampiran : -
Hal : Permohonan Ijin Penelitian
Kepada : **Kepala Klinik Kualita Medica Tegal**
Yth.

di tempat

Dengan hormat.

Mahasiswa dengan identitas berikut ini:

nama : Mukhammad Sabda Abdulloh
NIM : 16090106
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Tugas Akhir di Klinik Kualita Medica. Kami memohon Bapak memberikan izin kepada mahasiswa yang bersangkutan agar memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami sampaikan terima kasih.

Tegal, 21 April 2021
Ka. Prodi DIV Teknik Informatika,



Slamet Wiyono, S.Pd., M.Eng
NIPY 08.015.222