

**LAPORAN TUGAS AKHIR**

**PEMANFAATAN METODE CERTAINTY FACTOR**

**UNTUK DIAGNOSA PENYAKIT PASCA**

**MELAHIRKAN BERBASIS WEB**

***UTILIZATION OF CERTAINTY FACTOR METHODS***

***FOR WEB-BASED POSTPARTUM DISEASE***

***DIAGNOSIS***

Diajukan untuk memenuhi salah satu syarat mengerjakan dan menempuh ujian  
Tugas Akhir

**Bidang : Teknologi Web**



NPM : 1655201110009

Nama : Nia Sapitri

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH BANJARMASIN**

**BANJARMASIN**

**2020**

**LAPORAN TUGAS AKHIR**

**PEMANFAATAN METODE CERTAINTY FACTOR**

**UNTUK DIAGNOSA PENYAKIT PASCA**

**MELAHIRKAN BERBASIS WEB**

***UTILIZATION OF CERTAINTY FACTOR METHODS***

***FOR WEB-BASED POSTPARTUM DISEASE***

***DIAGNOSIS***

Diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik Informatika

**Bidang : *Teknologi Web***



NPM : 1655201110009

Nama : Nia Sapitri

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH BANJARMASIN**

**BANJARMASIN**

**2020**

## HALAMAN PERSETUJUAN TUGAS AKHIR

NPM : 1655201110009

Nama : Nia Sapitri

Program Studi : S1 Informatika

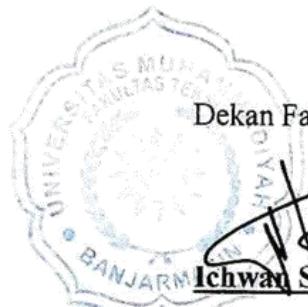
Fakultas : Teknik

Judul : Pemanfaatan Metode *Certainty Factor* Untuk Diagnosa Penyakit Pasca Melahirkan Berbasis Web

Tugas Akhir ini telah diperiksa dan disetujui,

Banjarmasin, 22 Juli 2020

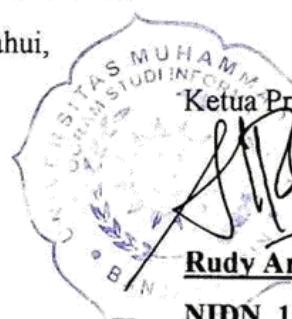
Mengetahui,



Dekan Fakultas Teknik

Ichwan Setiawan, M.T

NIDN. 1126078301



Ketua Program Studi Informatika

Rudy Ansari, M. Kom

NIDN. 1112068401

Koordinator Tugas Akhir

Kamarudin, M. Kom

NIDN. 1129097901

Menyetujui,

Pembimbing 1

Nahdi Saubari, M. Kom

NIDN. 1108109102

Pembimbing 2

Kamarudin, M. Kom

NIDN. 1129097901

## HALAMAN PENGESAHAN

NPM : 1655201110009

Nama : Nia Sapitri

Program Studi : S1 Informatika

Fakultas : Teknik

Judul : Pemanfaatan Metode *Certainty Factor* Untuk Diagnosa Penyakit Pasca Melahirkan Berbasis Web

Tugas Akhir ini telah diajukan dan dipertahankan dihadapan Tim Penguji pada Sidang tugas akhir tanggal 20 Juli 2020, Menurut pandangan kami, tugas akhir

Ini memadai dari segi kualitas maupun kuantitas untuk tujuan

**Penganugrahan gelar Sarjana Komputer (S. Kom)**

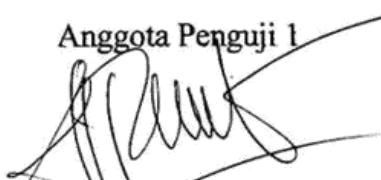
Banjarmasin, 22 Juli 2020

### Tim Penguji

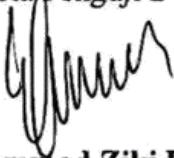
Ketua Penguji

  
Windarsyah, M.Kom  
NIDN. 1104078601

Anggota Penguji 1

  
Rudy Ansari, M.Kom  
NIDN. 1112068401

Anggota Penguji 2

  
Muhammad Ziki Elfirman, M. Kom  
NIDN. 1109098501

**PERNYATAAN**  
**KEASLIAN TUGAS AKHIR**

Sebagai mahasiswa Universitas Muhammadiyah Banjarmasin, yang bertanda tangan dibawah ini, saya :

Nama : Nia Sapitri  
NPM : 1655201110009

Menyatakan bahwa karya ilmiah saya yang berjudul :

Pemanfaatan Metode *Certainty Factor* Untuk Diagnosa Penyakit Pasca Melahirkan Berbasis Web

Merupakan karya asli saya (kecuali cuplikan dan ringkasan yang masing – masing telah saya jelaskan sumbernya dan perangkat pendukung dll). Apabila di kemudian hari, karya saya disinyalir bukan karya asli saya, yang disertai dengan bukti – bukti yang cukup, maka saya bersedia untuk dibatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Banjarmasin  
Pada tanggal : 22 Juli 2020

Yang menyatakan



( Nia Sapitri )

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Universitas Muhammadiyah Banjarmasin, yang bertanda tangan dibawah ini, saya :

Nama : Nia Sapitri

NPM : 1655201110009

Demi mengembangkan Ilmu Pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Banjarmasin Hak Bebas Royalti Non-Ekskusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul :

Pemanfaatan Metode *Certainty Factor* Untuk Diagnosa Penyakit Pasca Melahirkan Berbasis Web

Dengan Hak Bebas Royalti Non-Ekskusif ini Universitas Muhammadiyah Banjarmasin berhak untuk menyimpan, mengcopy ulang (memperbanyak), menggunakan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya dan menampilkan / mempublikasikannya diinternet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Muhammadiyah Banjarmasin, segala bentuk tuntunan hukum yang timbul atas pelanggaran Hak Cipta dan Karya Ilmiah saya ini.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Banjarmasin

Pada Tanggal : 22 Juli 2020

Yang menyatakan



## **UCAPAN TERIMAKASIH**

Dengan memanjatkan puji syukur kehadirat Allah S.W.T. Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya kepada penulis sehingga laporan tugas akhir dengan judul “PEMANFAATAN METODE CERTAINTY FACTOR UNTUK DIAGNOSA PENYAKIT PASCA MELAHIRKAN BERBASIS WEB” dapat penulis selesaikan dengan rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terimakasih kepada :

1. Prof. Dr. H. Ahmad Khaerudin, M.Ag. selaku Rektor Universitas Muhammadiyah Banjarmasin.
2. Ichwan Setiawan, M.T., selaku Dekan Fakultas Teknik.
3. Rudy Ansari, M. Kom., selaku Ketua Program Studi S1 Informatika.
4. Nahdi Saubari, M. Kom. selaku pembimbing I tugas akhir tang memberikan ide penelitian, memberikan informasi referensi yang penulis butuhkan dan bimbingan yang berkaitan dengan penelitian penulis.
5. Kamarudin, M. Kom selaku pembimbing II tugas akhir yang memberi ide penelitian, memberi informasi referensi yang penulis butuhkan dan bimbingan yang berkaitan dengan penelitian penulis.
6. Dosen – dosen pengampu di fakultas Teknik Universitas Muhammadiyah Banjarmasin yang telah memberikan ilmu dan pengalamannya masing – masing, sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.
7. Kepada orang tua bapak Masrani dan Fitriah yang selalu memberikan motivasi, dukungan, do'a dan banyak hal lain yang telah diberikan kepada penulis, sehingga penulis dapat menyelesaikan kuliah dan penelitian ini dengan baik.
8. Teman – teman dan sahabat yang tidak dapat penulis sebutkan satu – satu, yang banyak membantu, mendo'akan serta motivasi dan mendukung selama perkuliahan, sehingga penulis dapat menyelesaikan kuliahnya dengan baik.

Semoga Allah S.W.T. memberikan balasan yang lebih besar kepada beliau-beliau, dan pada akhirnya penulis berharap bahwa penulisan laporan tugas akhir ini dapat bermanfaat dan berguna sebagaimana fungsinya.

Banjarmasin, 22 Juli 2020

Penulis

## **ABSTRAK**

Kehamilan merupakan suatu perubahan besar bagi tubuh wanita, kehamilan akan mengakibatkan terjadinya perubahan seluruh sistem yang cukup mendasar. Melahirkan itu merupakan masa transmisi yang dianggap para wanita atau ibu yang dianggap menyenangkan. Meskipun demikian, masih banyak ibu yang hanya mengetahui tentang berbagai permasalahan ketika mereka sedang pada masa kehamilan, yang menyebabkan mereka tidak memikirkan ancaman penyakit pada saat pasca melahirkan. Solusi dari permasalahan tersebut yaitu diperlukan aplikasi diagnosa penyakit pasca melahirkan untuk mengetahui penyakit yang mungkin saja terjadi pada seorang ibu. Metode yang akan digunakan di penelitian ini yaitu dengan metode *Certainty Factor* agar dapat memudahkan para pakar untuk menghitung tingkat keyakinan user untuk mendiagnosa suatu penyakit pasca melahirkan yang terjadi pada ibu. Hasil dari laporan ini dapat membuktikan bahwa dengan menggunakan metode *Certainty Factor* ini mampu melakukan pendiagnosaan penyakit pasca melahirkan secara tepat sesuai dengan pendiagnosaan penyakit pasca melahirkan dari seorang pakar.

**Kata Kunci :** Pemanfaatan Metode *Certainty Factor* Untuk Diagnosa Penyakit Pasca Melahirkan

xiv + 99 halaman, 48 gambar, 31 tabel, 6 lampiran

Daftar acuan : 38 (2009 - 2020)

## DAFTAR ISI

HALAMAN PERSETUJUAN TUGAS AKHIR... <b>Error! Bookmark not defined.</b>	
HALAMAN PENGESAHAN.....	<b>Error! Bookmark not defined.</b>
PERNYATAAN.....	<b>Error! Bookmark not defined.</b>
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	iii
UCAPAN TERIMAKASIH.....	iv
ABSTRAK .....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan penelitian .....	2
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI .....	5
2.1 Tinjauan Studi .....	5
2.1.1 Sistem Pakar Diagnosa Penyakit Seorang Ibu Pasca Melahirkan .....	5
2.1.2 Implementasi Algoritma SAW ( <i>Simple Additive Weighting</i> ) <i>Dempster Shafer</i> pada Diagnosa Awal <i>Postpartum Depression</i> .....	5
2.1.3 Penerapan Metode <i>Depth First Seacrh</i> Pada Sistem Pakar Untuk Diagnosa Penyakit Pasca Melahirkan.....	6
2.1.4 Sistem Pakar Diagnosa Penyakit Kolesterol dan Asam Urat Menggunakan Metode <i>Certainty Factor</i> .....	7
2.2 Tinjauan Pustaka .....	8

2.2.1 Sistem Pakar .....	8
2.2.2 Diagnosa Penyakit .....	8
2.2.3 Pasca melahirkan .....	9
2.2.4 Pendarahan Postpartum .....	10
2.2.5 Infeksi Postpartum .....	10
2.2.5 Preeklamsia.....	10
2.2.6 Emboli Paru .....	11
2.2.7 Baby Blues.....	11
2.2.8 Metode Certainty Factor .....	11
2.2.9 PHP .....	12
2.2.10 Laravel .....	13
2.2.11 Laragon .....	14
2.2.12 Web.....	14
2.2.13 Visual Studio Code .....	15
2.2.14 Flowchart .....	15
2.2.15 UML ( <i>Unified Modeling Languege</i> ).....	18
2.2.16 Use Case Diagram .....	18
2.2.17 Aktivitas Diagram.....	19
2.2.18 Sequence Diagram .....	20
2.2.19 Class Diagram.....	23
2.3 Kerangka Pemikiran .....	24
<b>BAB III METODE PENELITIAN.....</b>	<b>29</b>
3.1 Metode Pengembangan Sistem .....	29
3.1.1 Analisa Kebutuhan.....	30
3.1.2 Desain Sistem .....	32

3.2 Instrumen Penelitian.....	58
BAB IV HASIL DAN PEMBAHASAN .....	60
4.1 Implementasi Perangkat Lunak .....	60
4.1.1 Implementasi Basis Data .....	60
4.1.2 Implementasi Interface .....	62
4.2 Pembahasan .....	95
BAB V KESIMPULAN DAN PENELITIAN SELANJUTNYA .....	97
5.1 Kesimpulan.....	97
5.2 Penelitian Selanjutnya .....	97
Jadwal Penyusunan Tugas Akhir .....	98
DAFTAR PUSTAKA .....	99

## **DAFTAR TABEL**

Tabel 2. 1 Simbol - Simbol Flowchart .....	15
Tabel 2. 2 Simbol-simbol yang ada pada diagram aktivitas. ....	20
Tabel 2. 3 Simbol-simbol yang ada pada sequence diagram. ....	21
Tabel 2. 4 Simbol-simbol yang ada pada class diagram. ....	23
Tabel 3. 1 Tabel Pertanyaan wawancara.....	31
Tabel 3. 2 Tabel Gejala. ....	32
Tabel 3. 3 Tabel Penyakit. ....	33
Tabel 3. 4 Daftar Hubungan Gejala dan Penyakit.....	33
Tabel 3. 5 Penyakit Preeklamsia .....	35
Tabel 3. 6 Penyakit Infeksi Postpartum. ....	35
Tabel 3. 7 Deskripsi Use Case. ....	43
Tabel 3. 8 Tabel User. ....	51
Tabel 3. 9 Tabel admin.....	51
Tabel 3. 10 Tabel Penyakit. ....	51
Tabel 3. 11 Tabel Gejala. ....	51
Tabel 3. 12 Tabel Basis Pengetahuan. ....	52
Tabel 3. 13 Tabel Kondisi.....	52
Tabel 3. 14 Tabel Hasil. ....	52
Tabel 3. 15 Tabel Post.....	52
Tabel 4. 1 Pengujian Black Box Menu Login Admin.....	63
Tabel 4. 2 Pengujian Black Box Menu Halaman Login User.....	65
Tabel 4. 3 Pengujian Black Box Menu Halaman Dashboard.....	68
Tabel 4. 4 Pengujian Black Box Menu Halaman User. ....	70
Tabel 4. 5 Pengujian Black Box Menu Halaman Penyakit.....	72
Tabel 4. 6 Pengujian Black Box Menu Halaman Gejala. ....	76
Tabel 4. 7 Pengujian Black Box Menu Halaman Pengetahuan. ....	80
Tabel 4. 8 Pengujian Black Box Menu Halaman Data User.....	83
Tabel 4. 9 Pengujian Black Box Menu Halaman Diagnosa Penyakit.....	87
Tabel 4. 10 Pengujian Halaman Untuk Memasukkan Data Penyakit. ....	89

Tabel 4. 11 Pengujian Halaman Untuk Memasukkan Data Gejala.....	91
Tabel 4. 12 Pengujian Halaman Untuk Memasukkan Data Pengetahuan.....	95

## **DAFTAR GAMBAR**

Gambar 2. 1 Logo PHP.....	13
Gambar 2. 2 Logo Laravel.....	14
Gambar 2. 3 Logo Laragon.....	14
Gambar 2. 4 Logo Visual Studio Code.....	15
Gambar 2. 5 Kerangka Pemikiran.....	25
Gambar 3. 1 Metode Pengebangan Sistem. ....	29
Gambar 3. 2 Flowchart Admin. ....	41
Gambar 3. 3 Flowchart User. ....	42
Gambar 3. 4 Use Case. ....	43
Gambar 3. 5 Class Diagram. ....	44
Gambar 3. 6 Sequence Diagram Admin Mengelola Data Penyakit.....	45
Gambar 3. 7 Sequence Diagram Melakukan Diagnosa Penyakit. ....	45
Gambar 3. 8 Sequence Diagram Admin Mengelola Data Pengetahuan. ....	46
Gambar 3. 9 Diagram Aktivitas Mengelola Data Penyakit.....	47
Gambar 3. 10 Aktivitas Diagram Mengelola Data Gejala. ....	48
Gambar 3. 11 Aktivitas Diagram Mengelola Data Pengetahuan. ....	49
Gambar 3. 12 Aktivitas Diagram Proses Diagnosa Penyakit.....	50
Gambar 3. 13 Relasi Antar Tabel.....	53
Gambar 3. 14 Desain Halaman Login Admin.....	54
Gambar 3. 15 16 Desain Halaman Login User. ....	54
Gambar 3. 16 Desain Halaman Dashboard. ....	55
Gambar 3. 17 Desain Halaman User.....	55
Gambar 3. 18 Desain Halaman Penyakit. ....	56
Gambar 3. 19 Desain Halaman Gejala. ....	56
Gambar 3. 20 Desain Halaman Pengetahuan. ....	57
Gambar 3. 21 Desain Halaman Diagnosa. ....	57
Gambar 3. 22 Desain Halaman Hasil Diagnosa. ....	58
Gambar 3. 23 Desain Halaman Hasil Cetak.....	58

Gambar 4. 1 Tabel Admin.....	60
Gambar 4. 2 Basis Pengetahuan.....	60
Gambar 4. 3 Tabel Gejala. ....	61
Gambar 4. 4 Tabel Penyakit.....	61
Gambar 4. 5 Tabel Kondisi. ....	61
Gambar 4. 6 Tabel Post.....	61
Gambar 4. 7 Tabel User. ....	62
Gambar 4. 8 Tampilan Halaman Login Admin. ....	63
Gambar 4. 9 Tampilan Halaman Login User. ....	65
Gambar 4. 10 Tampilan Halaman Dashboard.....	68
Gambar 4. 11 Tampilan Halaman User.....	69
Gambar 4. 12 Tampilan Halaman Penyakit. ....	72
Gambar 4. 13 Tampilan Halaman Gejala.....	75
Gambar 4. 14 Tampilan Halaman Pengetahuan.....	79
Gambar 4. 15 Tampilan Halaman Data User. ....	83
Gambar 4. 16 Tampilan Halaman Diagnosa Penyakit. ....	86
Gambar 4. 17 Tampilan Halaman Diagnosa Penyakit. ....	86
Gambar 4. 18 Tampilan Halaman Memasukkan Data Penyakit. ....	89
Gambar 4. 19 Tampilan Halaman Untuk Memasukkan Data Gejala. ....	91
Gambar 4. 20 Tampilan Halaman Untuk Memasukkan Data pengetahuan.....	94

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang Masalah**

Kehamilan merupakan suatu fase bagi tubuh wanita yang mengakibatkan terjadinya perubahan seluruh sistem yang cukup mendasar. Salah satu dari perubahan ini adalah tidak stabilnya kesehatan ibu melahirkan. Masih banyak ibu yang kurang mengetahui kesehatan dan penyakit yang mungkin saja bisa terjadi setelah melahirkan. Penyakit yang mungkin saja terjadi pada ibu pasca melahirkan yaitu pendarahan *Postpartum*, pendarahan *Postpartum* adalah pendarahan yang jumlahnya melebihi 500 ml dalam 24 jam pertama pasca melahirkan [2], infeksi *postpartum* adalah terjadi pada saat pasca melahirkan atau usai persalinan, ada kemungkinan terjadi robekkan pada jalan lahiratau perineum, luka terbuka pada saat rahim di tempat menempelnya plasenta, atau luka pada obrasi setelah obrasi ceaser [3], preeklamsia adalah komplikasi kehamilan serius yang ditandai dengan tekanan darah tinggi yang tidak terkontrol pada ibu hamil. [4], emboli paru adalah kondisi yang cukup berbahaya yang mengakibatkan oksegin terhambat, sehingga oksigen tidak mengalir pada tubuh dan merusak sistem tubuh. Emboli bisa menyerang ibu yang sedang dalam proses persalinan [5], penyakit *baby blues* merupakan kondisi yang dialami wanita berupa munculnya perasaan gundah dan sedih berlebihan [6].

Adapun gejala-gejalanya yang sering dialami ibu pasca melahirkan seperti darah yang keluar berwarna merah terang, turunnya tekanan darah, jantung berdebar, mau pingsan, dan lemas, sesak napas, konsentrasi menurun, urine keruh atau berdarah, merasa pusing cepat lelah, dan lain-lain. Sebagian besar ibu hanya mengetahui gejala-gejala umum yang terjadi dalam masa kehamilan saja, sehingga mereka melupakan bahwa adanya berbagai macam penyakit yang harus diperhatikan ketika mereka sudah melahirkan [1].

Pengetahuan penyakit setelah melahirkan dapat diperolah dari pakar atau ahli, dalam hal ini adalah dokter spesialis kandungan. Namun dokter spesialis

kandungan ini sulit untuk ditemui, karena jadwal yang padat untuk mengunjungi beberapa tempat proses ibu melahirkan dan bisa juga ke tempat ibu yang ingin memeriksakan kandungan, namun untuk bisa mendapatkan informasi dari dokter spesialis kandungan harus membuat janji terlebih dahulu. Sehingga apabila pakar atau doker kandungannya tidak ada, maka perawat atau petugasnya tidak harus menunggu pakarnya langsung untuk mengambil pengetahuan. Cara alternatif untuk mengecek dan mendapatkan informasi penyakit pasca melahirkan yang mudah dan cepat yaitu bisa dengan menggunakan aplikasi untuk mendiagnosa penyakit pasca melahirkan.

Kendala saat ini karena di Rumah Sakit Pembalah Batung Amuntai masih tidak adanya sistem yang dapat membantu untuk mendeteksi penyakit pasca melahirkan. ketidaktahuan ibu terhadap penyakit yang dialami pasca melahirkan dan ibu tidak bisa mendeteksi penyakit lebih dini, yang mungkin saja bisa terjadi, maka bisa menggunakan aplikasi diagnosa penyakit pasca melahirkan agar dapat mengukur atau mengetahui dengan cepat yang dialami ibu pasca melahirkan.

Agar dapat memperoleh keyakinan mendiagnosa penyakit setelah ibu melahirkan dan penanganan atau solusinya yaitu menggunakan metode *Certainty Factor* (CF). Metode ini merupakan perhitungan tingkat kepastian terhadap kesimpulan yang diperoleh dan dihitung berdasarkan nilai probabilitas penyakit karena adanya beberapa gejala [7]. Di mana aplikasi yang menggunakan metode ini mampu melakukan diagnosa sebuah penyakit setelah ibu melahirkan berdasarkan gejala-gelaja yang diderita oleh pasien dengan adanya gejala-gejala yang dialami ibu pasca melahirkan.

Aplikasi ini dibangun dengan berbasis web agar dapat diakses secara luas oleh pusat kesehatan untuk kelahiran, sehingga dapat membantu para medis terhadap ibu yang mengalami gejala-gejala pasca melahirkan untuk mengetahui penyakit apa yang dideritanya dan bagaimana cara penanganannya.

## **1.2 Rumusan Masalah**

Adapun untuk rumusan masalah diangkat dari judul penelitian, yaitu “Pemanfaatan Metode Certainty Factor untuk diagnosa penyakit pasca melahirkan” adalah bagaimana membuat suatu aplikasi yang berguna sebagai alat bantu bagi petugas dan perawat untuk mendiagnosa penyakit pasca melahirkan dengan metode *Certainty Factor* berbasis web dari data-data diagnosa penyakit pasca melahirkan?

## **1.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah ::

1. Penelitian yang dibangun hanya dilakukan pada Rumah Sakit Pembalahan Batung Amuntai yang menggunakan aplikasi ini hanya petugas dan perawat untuk mendapatkan data-data suatu gejala-gejala yang di alami pasca melahirkan, nama penyakit pasca melahirkan dan solusi atau cara penanganannya.
2. Dengan batasan jumlah penyakit ada 5 dan jumlah gejala ada 32.

## **1.4 Tujuan penelitian**

Adapun tujuan dari pembuatan aplikasi yang dibangun yaitu:

1. Memodelkan metode *Certainty Factor* untuk melakukan diagnosa penyakit berdasarkan gejala-gejala ibu melahirkan.
2. Membuat aplikasi yang berguna sebagai alat bantu dokter kandungan atau bidan untuk mendiagnosa penyakit pasca melahirkan dengan metode *Certainty Factor* berbasis web.

## **1.5 Manfaat Penelitian**

Manfaat dari penelitian yang dibangun adalah:

1. Manfaat bagi petugas atau perawat agar lebih cepat dalam mendiagnosa penyakit pasca melahirkan dan hasil diagnosa dapat langsung diberitahukan kepada dokter atau bidan dan dapat segera langsung ditangani.
2. Manfaat bagi ibu yaitu untuk mendapatkan informasi yang akurat dan ibu dapat merasa lebih tenang dan tidak stress yang dapat memperparah gejala atau penyakit.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Studi**

Penelitian dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Penelitian ini memperluas literature tentang keselarasan strategis. Penelitian yang dijadikan perbandingan tidak lepas dari topik penelitian.

##### **2.1.1 Sistem Pakar Diagnosa Penyakit Seorang Ibu Pasca Melahirkan**

Nanang Setiawan, Pulung, dan Nurtantion Andono, melakukan penelitian untuk mendiagnosa penyakit seorang ibu pasca melahirkan, dengan menggunakan metode *Forward Chaining* dan *Certainty Factor* akan menghasilkan sebuah sistem pakar untuk diagnosa penyakit seorang ibu pasca melahirkan, melahirkan dan menjadi ibu adalah fisiologis wanita, dan ini merupakan masa transmisi yang mereka anggap menyenangkan. Tetapi, banyak ibu yang hanya mengetahui tentang permasalahan ketika mereka sedang dengan masa kehamilan saja, sehingga mereka tidak memikirkan ancaman penyakit yang sering terjadi setelah melakukan persalinan dan akhirnya mereka mendapat masalah dalam hal adanya penyakit pasca melahirkan [1].

Pengembangan program dan analisis data agar dapat lebih diperluas cakupannya sesuai dengan kebutuhan program, dan sistem yang dibangun masih banyak kekurangan, baik dari segi fungsionalitas maupun data yang dimiliki. Oleh karena itu, sangat dibutuhkan berbagai pengembangan lebih lanjut agar dapat memberikan lebih banyak lagi manfaat bagi masyarakat luas [1].

##### **2.1.2 Implementasi Algoritma SAW (*Simple Additive Weighting*) Dempster**

###### ***Shafer pada Diagnosa Awal Postpartum Depression***

Yuli Kartika Sari, Dwi Kartini, dan Muliadi Melakukan penelitian dengan menggunakan algoritma SAW (*Simple Additive Weighting*) dan *Dempster Shafer*. Depresi *Postpatum* (*Postpartum Depression*) merupakan salah satu bentuk depresi yang dialami ibu setelah melahirkan bayi pertama dan berlangsung pada tahun

pertama setelah kelahiran bayi. Ini disebabkan karena periode transisi kehidupan baru yang cukup membuat stress namun tidak semua ibu yang mampu melakukan adaptasi dan mengatasi stresor tersebut timbul keluhan-keluhan antara lain berupa stress, cemas dan depresi [8].

Penelitian ini menggabungkan dua metode, yaitu metode *Simple Additive Weighting* dan *Dempster Shafer*. Metode ini digunakan untuk melakukan diagnosa awal berdasarkan gejala-gejala yang telah diinformasikan penderita depresi *Postpartum* berdasarkan basis pengetahuan yang diperoleh dari pakar, seorang psikolog, dan bidan yang direpresentasikan ke dalam sistem pakar. Input yang digunakan berupa beberapa gejala yang dialami dan dirasakan oleh pasien pasca melahirkan yang akan dikonversikan menjadi nilai 1 kemudian diperoleh menggunakan metode *Simple Additive Weighting* untuk menentukan nilai belief gejala. Nilai belief gejala tersebut digunakan untuk melakukan diagnosa awal *Postpartum Depression* dengan metode *Dempster Shafer*. Dan hasil implementasi diagnose gangguan postpartum Depression yang telah diujikan pada sistem pakar berdasarkan data rekam medik 90% [8].

Dari penelitian yang telah dilakukan, diharapkan peneliti selanjutnya juga dapat menelaah dan mengkaji dengan lebih dalam mengenai sub gejala dan gejala pada bidang psikiologi karena terdapat banyak sub gejala yang samar [8].

#### 2.1.3 Penerapan Metode *Depth First Seacrh* Pada Sistem Pakar Untuk Diagnosa Penyakit Pasca Melahirkan

Bahar, Andri Suseno, melakukan penelitian untuk membuat sistem pakar diagnosa penyakit pasca melahirkan. Sistem dibuat agar orang awam dapat menyelesaikan masalahnya yang rumit yang hanya dapat diselesaikan dengan bantuan para ahli pada bidang tertentu. Agar sistem dapat menyelesaikan masalah tersebut diperlukannya sebuah alur sistem, dimana alur sistemnya tersebut dapat mencari solusi dari permasalahan seperti yang biasa dilakukan oleh seorang pakar, maka dibutuhkan pendekatan dengan menggunakan metode *Depth First* untuk mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli [9].

Sistem yang berasal dari pikiran pakar, semua data benar-benar berasal dari pakar dalam bidangnya, dan bukan dari seorang sembarangan yang dibuat agar orang awam dapat menyelesaikan masalahnya yang rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Sistem pakar ini akan dapat membantu aktivitasnya sebagai asisten yang sangat berpengalaman [9].

Dari penelitian ini, peneliti dapat memberikan saran untuk penelitian selanjutnya agar dapat mencoba metode pembobotan yang lain guna endapatkan bobot yang benar-benar yang dapat mempresentasikan pengetahuan pakar [9].

#### 2.1.4 Sistem Pakar Diagnosa Penyakit Kolesterol dan Asam Urat Menggunakan Metode *Certainty Factor*

Patmawati Hasan, Eka Wahyu Sholeha, Yuslius Nahak Tetik, dan Kusrini, melakukan sebuah penelitian tentang sistem pakar diagnose penyakit kolesterol dan asam urat menggunakan metode *Certainty Factor*. Kolestrol dan asam urat merupakan penyakit yang mempunyai tingkat suatu kejadiannya yang cukup tinggi. Berdasarkan dari seorang ahli dari *Clinical Research Support Unit (CRSU)* Fakultas Kedokteran Universitas Indonesia, DR. Nafrialdi, PhD menyatakan bahwa penduduk Indonesia memiliki kadar kolesterol lebih tinggi dari batas normal dan menurut data WHO 2015, penderita asam urat di Indonesia terjadi pada usia dibawah 24 tahun sebesar 32% dan di atas 34 tahun sebesar 68%.

Namun masih banyak ketidaktahuan masyarakat umum terhadap penyakit yang diderita di karenakan mahalnya biaya yang harus dikeluarkan agar dapat mengetahui suatu penyakit yang diderita lebih dini tanpa harus berkonsultasi ke dokter. Untuk membantu mengatasi permasalahan tersebut maka dibuatlah suatu aplikasi sistem pakar yang dapat mengidentifikasi penyakit kolesterol dan asam urat untuk masyarakat umum.

Metode yang digunakan untuk aplikasi sistem pakar diagnosa penyakit kelosterol dan asam urat menggunakan metode *Certainty Factor* untuk menyatakan kepercayaan dalam sebuah kejadian fakta atau hipotesis berdasarkan bukti atau

penilaian seorang pakar. Rekapitulasi Validasi Sistem melalui pakar memberikan keakuratan 80% terhadap sistem pakar tersebut [7] .

Diharapkan nantinya dapat mencakup banyak gejala yang lebih spesifikasi dan sistem juga dapat memberikan solusi pada tiap penyakit. Sehingga sistem tidak hanya mendiagnosis secara dini penyakit kolesterol dan asam urat saja [7].

## 2.2 Tinjauan Pustaka

### 2.2.1 Sistem Pakar

Menurut Turban Sistem pakar adalah sistem yang merupakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan seorang pakar atau seorang ahli pakar [10].

Sistem pakar (*expert system*) merupakan cabang dari kecerdasan buatan dan juga merupakan bidang ilmu yang muncul seiring dengan perkembangan ilmu komputer saat ini. Sistem ini adalah sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar, sistem ini bekerja untuk mengadopsi pengetahuan manusia kekomputer yang menggabungkan dasar pengetahuan (*knowledge base*) dengan sistem inferensi untuk mengantikan fungsi seorang pakar dalam menyelesaikan suatu masalah [11].

### 2.2.2 Diagnosa Penyakit

Secara etimologi, diagnosa berasal dari Bahasa Yunani dari kata Gnosis berarti ilmu pengetahuan. Jadi pengertian diagnosa secara teminologi ialah penetapan suatu keadaan yang menyimpang atau keadaan normal melalui dasar pemikiran dan pertimbangan ilmu pengetahuan. Diagnosa merupakan pendekatan sistematis terhadap pemahaman dan gambaran kondisi terkini organisasi yang merinci pada hakekat permasalahan dan identifikasi faktor penyebab yang memberikan dasar untuk pilih strategi perubahan dan teknik yang paling tepat [12].

Menurut kamus besar Bahasa Indonesia diagnosa merupakan penentuan untuk jenis penyakit dengan cara meneliti atau memeriksa suatu gejala-gejala yang dialami seseorang. Dan didalam konsep diagnosa telah tercakup pula konsep

prognosisnya, didalam proses diagnosa bukan hanya sekedar mengidentifikasi suatu jenis dan karakteristiknya, serta latar belakang dari suatu kelemahan penyakit tertentu yang melainkan mengimplikasikan suatu upaya untuk meramalkan suatu kemungkinan dan menyarankan tindakan pemecahannya [11].

Penyakit merupakan kejadian yang tidak normal pada tubuh yang tidak normal pada tubuh atau pikiran yang menyebabkan ketidaknyamanan. Menurut para ahli Dr. Eko Dudiarto penyakit merupakan kegagalan proses adaptasi suatu makhluk hidup bereaksi secara tepat terhadap rangsangan atau tekanan sehingga muncul gangguan pada fungsi atau morfologi organ atau sistem tubuh.

Dan menurut Kathleen Meehan Arias penyakit merupakan suatu kesakitan yang biasanya memiliki sedikitnya dua sifat dari kriteria ini: agen atiologik telah diketahui, golongan tanda seta gejala yang dapat diidentifikasi, atau perubahan anatomi yang tetap [13] .

### 2.2.3 Pasca melahirkan

Pasca melahirkan adalah keadaan masa sesudah melahirkan atau persalinan. Masa beberapa jam sesudah lahirnya plasenta atau tali pusat sampai minggu keenam setelah melahirkan. Masa melahirkan dimulai setelah kelahiran plasenta dan berakhir ketika alat-alat kandungan kembali pada masa sebelum hamil yang berlangsung kira-kira enam minggu, setelah melahirkan yang meliputi minggu-minggu berikutnya pada waktu saluran reproduksi kembali keadaan yang normal pada saat sebelum hamil [14].

Melahirkan adalah proses yang fisiologis dan merupakan kejadian yang menakjubkan bagi seorang ibu dan keluarga [15]. Melahirkan merupakan proses menegangkan, namun setelah itu terasa lega dan senang pada ibu dan keluarga. Tetapi proses terus berlanjut selama beberapa minggu setelah melahirkan, karena tubuh melakukan pemulihan dan penyesuaikan diri dengan kondisinya yang baru. Setelah melahirkan, tubuh masih melakukan berbagai perubahan [16].

#### 2.2.4 Pendarahan Postpartum

Pendarahan *postpartum* adalah pendarahan hebat yang terjadi pasca melahirkan. Penyebab paling umum dari pendarahan *postpartum* ini yaitu, terbukanya pembuluh darah pada rahim. Biasanya, pembuluh darah tersebut mengalami tekanan kontraksi rahim, yang terjadi setelah melahirkan [17].

Pendarahan *postpartum* adalah pendarahan yang jumlahnya melebihi 500ml dalam 24 jam pertama pasca melahirkan. Diperkirakan bahwa pendarahan *postpartum* terjadi sekitar 5% dari semua persalinan pervagina, dan sekitar seperempat dari seluruh kematian ibu disebabkan oleh pendarahan *postpartum* [2].

#### 2.2.5 Infeksi Postpartum

Infeksi *postpartum* yaitu terjadi pada saat pasca melahirkan atau usai persalinan, ada kemungkinan terjadi robekan pada jalan lahir atau perineum, luka terbuka pada rahim di tempat menempelnya plasenta, atau luka pada operasi setelah oprasi ceaser. Daerah yang mengalami luka tersebut menjadi rentan terhadap infeksi. Infeksi *postpartum* pada rahim meningkatkan resiko terjadinya sumbatan pembuluh darah, jika infeksi dibiarkan, maka bakteri dapat masuk ke dalam peredaran darah yang bisa menyebabkan sepsis [3]. Sepsis adalah komplikasi serius berbahaya dari suatu infeksi. Sepsis muncul ketika senyawa kimia yang disalurkan ke dalam aliran darah untuk melawan infeksi menyebabkan radang dan pembengkakan di berbagai bagian tubuh [18].

#### 2.2.5 Preeklamsia

Preeklamsia adalah komplikasi kehamilan serius yang ditantai dengan tekanan darah tinggi yang tidak terkontrol pada ibu hamil. Preeklamsia adalah penyebab kematian utama para ibu di negara-negara berkembang [19].

Faktor resiko preeklamsi lebih banyak terjadi pada primigravida, nullipara, pada usia ibu kurang dari 25 tahun atau lebih dari 35 tahun, kejadian meningkat dengan makin tuanya umur kehamilan. Usia ibu hamil kurang dari 25 tahun atau lebih dari 35 tahun berkaitan erat dengan berbagai komplikasi yang terjadi selama kehamilan, persalinan, nifas dan juga kesehatan bayi ketika masih dalam

kandungan maupun setelah melahirkan. Komplikasi adalah kesakitan pada ibu hamil, ibu bersalin, ibu nifas yang dapat mengancam jiwa ibu atau bayi [4].

### 2.2.6 Emboli Paru

Emboli paru adalah konsisi yang cukup berbahaya yang mengakibatkan oksigen terhambat, sehingga oksigen tidak mengalir pada tubuh dan merusak sistem tubuh. Emboli bisa menyerak ibu yang sedang dalam proses persalinan dan dikenal dengan emboli air ketuban [5].

Emboli air ketuban terjadi saat masuknya cairan ketuban dan komponennya ke dalam jaringan pembuluh darah. Diagnosis ini mengakibatkan oksigen ke dalam tubuh ibu menjadi terhambat. Jika tidak segera ditangani, emboli dapat membuat ibu memiliki masalah kesehatan ke depannya, seperti gangguan saraf [5].

### 2.2.7 Baby Blues

*Baby blues* adalah suatu gangguan psikologis sementara yang ditandai dengan memuncaknya emosi pada minggu pertama pasca melahirkan. *Baby blues* adalah sebuah perasaan sedih tanpa dasar terjadi setelah melahirkan [20].

*Baby blues* merupakan kondisi yang dialami wanita berupa munculnya perasaan gundah dan sedih berlebihan. *Baby blues* akan memburuk pada 3-4 hari pasca melahirkan kondisi ini juga biasanya hanya terjadi pada 14 hari pertama [6].

### 2.2.8 Metode Certainty Factor

*Certainty Factor* diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN [21]. *Certainty Factor* merupakan metode yang mengukur nilai kepastian yang diberikan oleh pakar terhadap suatu aturan dan mengatasi kesulitan dalam menentukan suatu gejala-gejala terhadap penyakit yang di alami seseorang [11]. *Certainty Factor* menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar [7].

Berikut notasi fakta kepastian:

$$MB[h, e^\Delta e2] = MB[h, e1] + MB[h, e2]x(1 - MB[h, e1]) \dots \dots \dots (3)$$

## Keterangan:

$CF(h,e)$  : *Certainty Factor* dari hipotesis h yang dipengaruhi oleh gejala (evidence) e. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB(h,e)$  : ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesa  $h$  yang dipengaruhi oleh gejala  $e$ .

$MD(h,e)$  : ukuran kenaikan ketidakpastian (*measure of increased disbelief*) terhadap hipotesis  $h$  yang dipengaruhi oleh gejala  $e$ .

**h** : Hipotesa (antara 0 dan 1).

E : Peristiwa / fakta (evidence).

## 2.2.9 PHP

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995 dan terus dikembangkan hingga saat ini. Ada banyak sekali web termasuk CMS yang dibuat menggunakan 12agno PHP, seperti WordPress dan lain-lain.

PHP atau *Hypertext Preprocessor* adalah linguagem de programação *script server side* que foi criada para gerar conteúdos web dinâmicos. PHP é uma linguagem de programação orientada a objetos que pode ser integrada ao HTML. PHP é uma linguagem de programação que pode ser executada no lado do servidor, permitindo que os desenvolvedores criem páginas web interativas e dinâmicas.

PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi Bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana [24].



Gambar 2. 1 Logo PHP.

Contoh kode PHP yaitu:

- Input

```
>?php
```

```
    echo "Hallo Dunia";
```

```
?>
```

- output

```
    Hallo Dunia.
```

#### 2.2.10 Laravel

Laravel adalah salah satu framework PHP terbaik yang dikembangkan oleh *Taylor Otwell*, projek Laravel dimulai dari pada April 2011. Awal mula, projek ini dibuat, karena *Otwell* sendiri tidak menemukan framework yang *up-to-date* dengan versi PHP. Mengembangkan framework yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa tahap keterbatasan tersebut, *Otwell* membuat sendiri framework dengan nama Laravel [22].

Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. Laravel berada di bawah lisensi *MIT License* dengan menggunakan Github sebagai tempat berbagai *code* untuk menjalankannya. Laravel adalah pengembangan *website* berbasis MVP (*Model View Controller*) yang ditulis dalam PHP yang dirancak untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu [25] [26].



Gambar 2. 2 Logo Laravel.

### 2.2.11 Laragon

Laragon adalah *Universal Development Environment* untuk PHP, Node.js, Python, Java, Go, Ruby, yang *portable*, terisolasi, cepat, ringan, dan mudah dipakai. Perangkat lunak sama halnya dengan XAMPP dan WAMP, yang digunakan untuk membangun local development environment pada sistem operasi windows. Namun selain menyertakan PHP, Apache Web Server, dan Database MySQL dalam paket instalasinya, laragon juga menyediakan segudang fitur menarik yang sangat relevan dengan kebutuhan pengembangan web [27] [28]. Laragon adalah perangkat lunak bebas yang mendukung banyak sistem operasi, berfungsi sebagai server diri sendiri/localhost.



Gambar 2. 3 Logo Laragon.

### 2.2.12 Web

Website adalah keseluruhan halaman-halaman web yang terdapat yang terdapat dalam sebuah domain yang mengandung ruang informasi. Dengan menggunakan teknologi hypertext, pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen web yang ditampilkan dalam browser web. Web memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri informasi di internet [29] [30] [31].

### 2.2.13 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung 15+ bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace visual studio code seperti C++, C#, Python, Go, Java) [32].



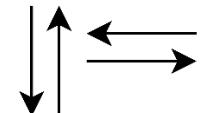
Gambar 2. 4 Logo Visual Studio Code.

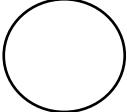
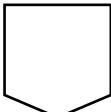
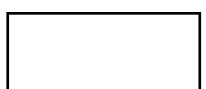
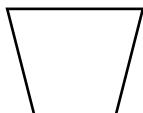
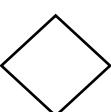
### 2.2.14 Flowchart

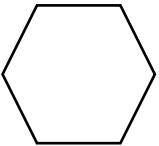
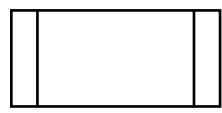
*Flowchart* menunjukkan gambaran bisnis proses yang merupakan kumpulan proses yang berisi kumpulan aktivitas terstruktur dan saling berelasi satu sama lain untuk menghasilkan keluaran bisnis proses sistem pakar diagnose penyakit pada pasca melahirkan [33].

Berikut adalah simbol - simbol yang ada pada *flowchart* kelas [34]:

Tabel 2. 1 Simbol - Simbol Flowchart.

Simbol	Deskripsi
Flow Direction Symbol 	Flow direction symbol yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan yang lain. Simbol ini disebut juga connecting line.
Terminator Symbol 	Terminator symbol yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan.

Connector Symbol 	Connector symbol yaitu simbol untuk keluar-masuk atau penyambungan proses dalam lembar atau halaman yang sama
Connector Symbol 	Connector symbol yaitu simbol untuk keluar-masuk atau penyambungan proses pada lembar atau halaman yang berbeda.
Processing Symbol 	Processing symbol yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
Symbol Manual Operation 	Symbol manual operation yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
Symbol Decision 	Symbol decision yaitu simbol pemilihan proses berdasarkan kondisi yang ada.
Symbol Input-Output 	Symbol input-output yaitu simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
Symbol Manual Input	Symbol manual input yaitu simbol untuk pemasukkan data secara manual on-line keyboard.

	
<p>Symbol Preparation</p> 	Symbol preparation yaitu simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
<p>Symbol Pradefine Proses</p> 	Symbol Pradefine proses yaitu simbol untuk pelaksanaan suatu bagian (sub-program) atau procedure.
<p>Symbol Display</p> 	Symbol display yaitu simbol yang menyatakan peralatan output yang digunakan yaitu layer, plotter, printer dan sebagainya.
<p>Symbol disk and On-line Storage</p> 	Symbol disk and on-line storage yaitu simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
<p>Symbol magnetic tape</p> 	Symbol magnetic tape unit yaitu simbol yang menyatakan input berasal dari pita diagnose atau output disimpan ke pita magnetic.
<p>Symbol Punch Card</p>	Symbol punch card yaitu simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu.

	
<p>Symbol Dokumen</p> 	Symbol dokumen yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

### 2.2.15 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah salah satu standar Bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [35].

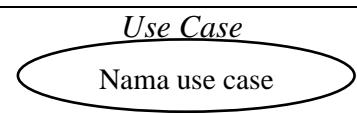
### 2.2.16 Use Case Diagram

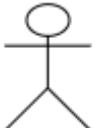
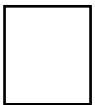
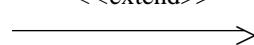
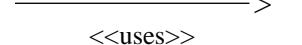
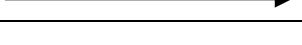
use case merupakan pemodelan untuk melakukan (*behaviour*) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Secara kasar, Use case digunakan untuk mengetahui fungsi apa saja ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [35].

Use case merupakan deskripsi lengkap tentang bagaimana sistem atau perangkat lunak berperilaku untuk para *actor*-nya. Dengan demikian Use case diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para *actor* dengan sistem atau perangkat lunak yang sedang dikembangkan [36].

Berikut ini adalah symbol – symbol yang ada pada diagram *Use Case* :

Tabel 2. 2 Simbol-simbol yang ada pada diagram Use Case.

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan system sebagian unit-unit yang saling bertukarkan pesan antar unit atau <i>actor</i> ; biasanya dinyatakan dengan

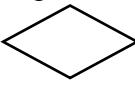
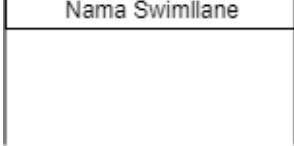
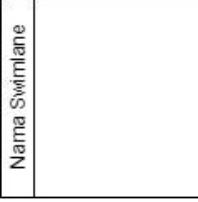
	menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
Aktor /actor  Nama actor	Orang, proses, atau system lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
System boundary 	Mensifikasikan paket yang menampilkan paket secara terbatas
Asosiasi /association 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / extend 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <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.
Generalisasi/ generalization 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / include/ user  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

### 2.2.17 Aktivitas Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem [35].

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2. 2 Simbol-simbol yang ada pada diagram aktivitas.

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

#### 2.2.18 Sequence Diagram

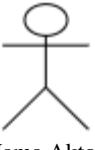
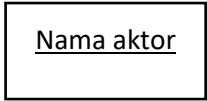
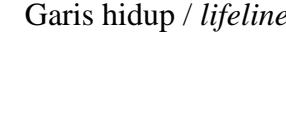
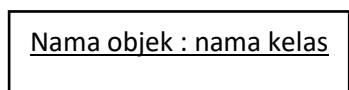
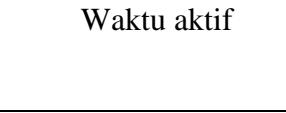
Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima

antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang telibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat diagnosis yang ada pada *use case*.

Banyaknya diagram sekuen yang harus Digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang mendefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak [35].

Berikut adalah simbol-simbol yang ada pada *sequence diagram*:

Tabel 2. 3 Simbol-simbol yang ada pada sequence diagram.

Simbol	Deskripsi
Actor  Nama Aktor Atau 	Orang,proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Tanpa waktu aktif 	
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah

	<p>tahapan yang dilakukan di dalamnya, misalnya</p> <pre> sequenceDiagram     participant Actor     participant Objek     Actor-&gt;&gt;Objek: 1: login()     activate Objek     Objek-&gt;&gt;Objek: 2: CekStatusLogin()     deactivate Objek     Objek-&gt;&gt;Actor: 3: open()   </pre> <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login() Actor tidak memiliki waktu aktif</p>
Pesan tipe create  <<create>>	<p>Menyatakan suatu objek memuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
Pesan tipe call  1: nam_metode()	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri ,</p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
Pesan tipe send  1: masukan	<p>Menyatakan bahwa suatu objek mengirimkan data /masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
Pesan tipe return	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian</p>

<p>1: keluaran</p>	ke objek tertentu, arah panas mengarah pada yang menerima kembalian
<p>Pesan tipe destroy</p>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

### 2.2.19 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut metode atau oprasi.

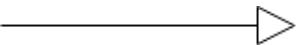
- atribut merupakan variable variable yang dimiliki oleh suatu kelas.
- oprasi atau merode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan dan perangkat lunak sinkron [35].

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2. 4 Simbol-simbol yang ada pada class diagram.

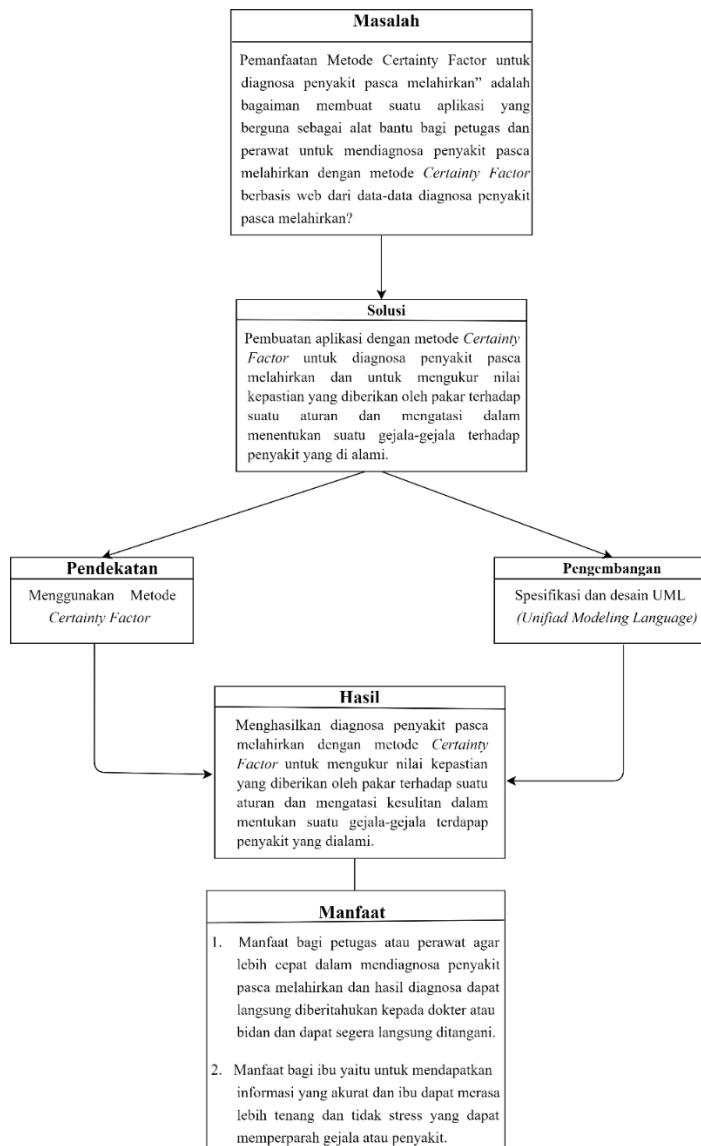
Simbol	Diskripsi
Kelas 	Kelas pasa struktur sistem

Antarmuka / <i>assocoantion</i>   Nama_Interface	Sama dengan konsep interface dalam pemograman berorientasi objek.
Asosiasi / <i>associantion</i>  	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i>  	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Generalisasi  	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i>  	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/ <i>aggregation</i>  	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

### 2.3 Kerangka Pemikiran

Kerangka pemikiran adalah model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah penting. Pada pembuatan aplikasi dibutuhkan beberapa tahapan yang harus dilalui

untuk dapat menghasilkan aplikasi pemanfaatan metode *certainty factor* untuk diagnosa penyakit pasca melahirkan yang berguna untuk tempat kesehatan tepatnya pada tempat dokter kandungan atau bidan.



Gambar 2. 5 Kerangka Pemikiran.

Penjelasan pada gambar diagram kerangka pemikiran diatas yaitu:

#### 1. Permasalahan

Permasalahan, didalam kerangka pemikiran yaitu:

“Pemanfaatan Metode Certainty Factor untuk diagnosa penyakit pasca melahirkan” adalah bagaimana membuat suatu aplikasi yang berguna sebagai alat bantu bagi petugas dan perawat untuk mendiagnosa penyakit pasca melahirkan dengan metode *Certainty Factor* berbasis web dari data-data diagnosa penyakit pasca melahirkan?

#### 2. Solusi

Solusi dari masalah-masalah yang ada, solusinya yaitu membuat suatu aplikasi dengan metode *Certainty Factor* untuk diagnosa penyakit pasca melahirkan.

#### 3. Pendekatan

Masalah dan solusinya maka diagnosa suatu pendekatan untuk menyelesaikan pembuatan aplikasi diagnosa penyakit pasca melahirkan yaitu dengan menggunakan metode *Certainty Factor*.

#### 4. Pengembangan

Masalah, solusi, dan adanya pendekatan menggunakan metode *Certainty Factor*, selanjutnya ada pendekatan yaitu untuk spesifikasi dan design menggunakan UML (*Unified Modeling Language* [35]).

#### 5. Hasil

Hasil dari beberapa masalah, solusi, pendekatan, dan pengembangan tersebut maka, hasil yang akan di dapat yaitu suatu aplikasi diagnosa penyakit pasca melahirkan dengan metode *Certainty Factor*.

## 6. Manfaat

Masalah, solusi, pendekatan, dan pengembangan yang digunakan, dan hasil yang didapatkan maka, ada beberapa manfaat yang di dapatkan dari aplikasi diagnosa penyakit pasca melahirkan yaitu:

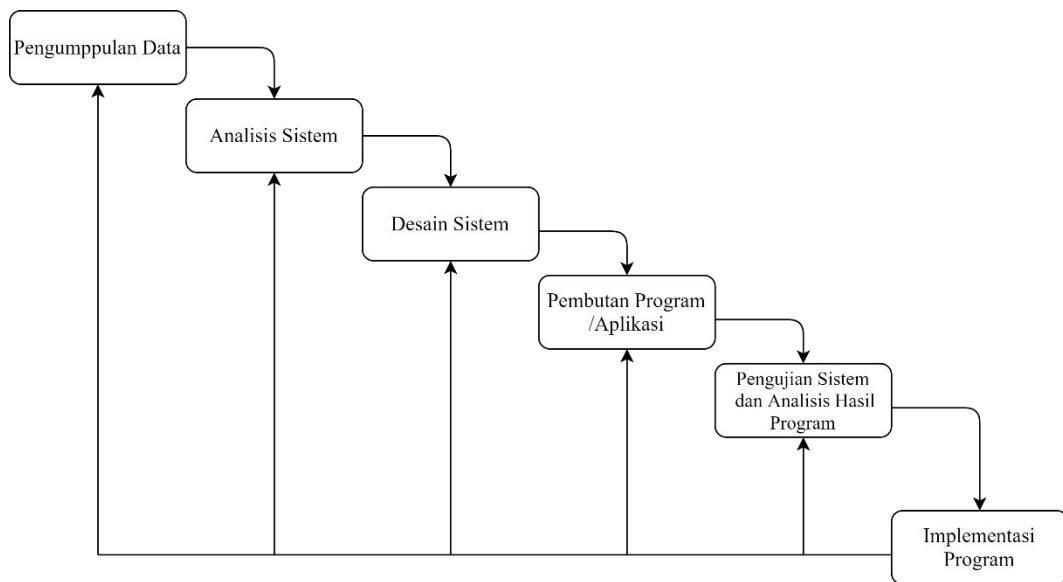
1. Manfaat bagi petugas atau perawat agar lebih cepat dalam mendiagnosa penyakit pasca melahirkan dan hasil diagnosa dapat langsung diberitahukan kepada dokter atau bidan dan dapat segera langsung ditangani.
2. Manfaat bagi ibu yaitu untuk mendapatkan informasi yang akurat dan ibu dapat merasa lebih tenang dan tidak stress yang dapat memperparah gejala atau penyakit.

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Pengembangan Sistem

Penelitian yang akan dilakukan yaitu, memalui tahapan-tahapan SDLC (*Systems Development Life Cycle*) dengan model penelitian yang diterapkan pada penelitian ini adalah dengan metode *waterfall*. Metode *waterfall* merupakan proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir kebawah melewati fase-fase [37] dari pengumpulan data, analisis sistem, desain sistem, pembuatan program atau aplikasi, pengujian sistem dan analisis hasil program dan implementasi program. Adapun pengembangan sistem model *waterfall* dapat dilihat pada gambar 3.1.



Gambar 3. 1 Metode Pengembangan Sistem.

Pada gambar 3.1 tahap awal yang dilakukan yaitu mengumpulkan data-data serta pengambilan data-data yang ada untuk dijadikan bahan penelitian. Selanjutnya melakukan analisis sistem, pada tahap analisis sistem ini dilakukan analisis terhadap sistem yang akan dibangun. Analisis tersebut meliputi Analisa input dan output sistem. Hasil analisis dapat dibuat dalam bentuk *Flowchart*, *Use Case*, *Aktivity Diagram*, dan *Sequence Diagram*. Selanjutnya tahap desain sistem, yaitu

tahapan membuat rancangan awal dari sistem yang akan dibangun misal, database serta desain interface. Kemudian tahap selanjutnya yaitu tahap pembuatan program atau aplikasi yang akan dibangun berbasis web dengan menggunakan Laravel yaitu salah satu framework berbasis PHP. Selanjutnya pengujian sistem dan analisis dari hasil program yang sudah dirancang. Setelah melalui beberapa tahapan dari pengumpulan data, analisis sistem, desain sistem, pembuatan program atau aplikasi, serta pengujian sistem dan analisis hasil program, selanjutnya adalah imlementasi program.

### 3.1.1 Analisa Kebutuhan

Pengumpulan data yang dilakukan yaitu sebagai berikut:

#### a. Observasi

Dalam objek penelitian dilakukan prosedur awal peneliti mendatangi pihak administrasi Fakultas Teknik untuk meminta surat permohonan meminta data penyakit pasca melahirkan di Rumah Sakit Pembalah Batung Amuntai. Selanjutnya peneliti pergi ke Rumah Sakit Pembalah Batung Amuntai untuk menemui pihak tata usaha yang ada di Rumah Sakit Pembalah Batung Amuntai dan menyerahkan surat permohonan tersebut kepada pihak tata usaha Rumah Sakit. Setelah surat tersebut diserahkan diminta dari pihak tata usahanya untuk memberikan nomor telepon agar bisa dihubungi jika surat yang peneliti ajukan sudah ditanda tangani atau diijinkan oleh pihak direktur Rumah Sakit Pembalah Batung Amuntai untuk bisa mendapatkan data-data penyakit pasca melahirkan.

Setelah satu minggu menunggu akhirnya pihak tata usaha Rumah Sakit memberi informasi kepada peneliti bahwa surat tersebut bahwa surat sudah ditanda tangani atau sudah diperbolehkan untuk melakukan penelitian oleh direktur Rumah Sakit Pembalah Batung Amuntai. Lalu peneliti menemui pihak tata usaha Rumah Sakit Pembalah Batung Amuntai untuk mengambil surat tersebut, dan diarahkan oleh pihak tata usaha untuk bisa langsung datang ke bagian bidang yang ingin diminta

data. Dan setelah itu peneliti mendatangi ke bagian bidang yang ingin diminta data yaitu di bagian ke bidanan.

b. Studi Pustaka

Pada tahap studi pustaka penelitian yang akan dilakukan yaitu dengan datang keperpustakaan untuk mendapatkan referensi terkait materi ataupun bahan pustaka lainnya. Dan ada beberapa referensi materi yang saya ambil dari buku yang saya punya yaitu ada buku rekayasa perangkat lunak, php untuk programmer pemula, dan mudah menggunakan *Framework* Laravel.

c. Wawancara

Pada tahap ini melakukan wawancara untuk mendapatkan pengumpulan data-data penyakit dan gejala penyakit bertemu dengan ibu Hj.Sri Suwarni,Amd.Keb dan ada beberapa bidan yang ada ruang kebidanan.

Pertanyaan-pertanyaan pada saat wawancara yang ditanyakan yaitu:

Tabel 3. 1 Tabel Pertanyaan wawancara.

No	Pertanyaan
1.	Apakah di rumah sakit Pembalih Batung Amuntai ada pasien yang mengalami penyakit pasca melahirkan?
2.	Apakah banyak pasien yang mengalami penyakit pasca melahirkan?
3.	Apa-apa saja nama-nama penyakit yang di alami pasca melahirkan?
4.	Apa-apa saja gejala-gejala penyakit yang terjadi?

### 3.1.2 Desain Sistem

#### a. Perhitungan Algoritma

Bagian ini menjelaskan langkah-langkah atau cara untuk menguji metode yang diusulkan dalam penelitian ini.

##### 1. Basis Pengetahuan

Keberhasilan suatu aplikasi sistem pakar terletak pada basis pengetahuan dan bagaimana mengolah pengetahuan tersebut agar dapat bisa ditarik sebagai kesimpulan. Tabel penyakit dan tabel gejala tersebut digunakan sebagai pola pencocokan informasi yang dimasukkan oleh pengguna.

Tabel 3. 2 Tabel Gejala.

Kode Gejala	Gejala
G01	Perdarahan hebat yang terus menerus keluar dari vagina setelah persalinan
G02	Darah yang keluar berwarna merah terang
G03	keluarnya bekuan darah yang berukuran lebih besar
G04	Cepat Lelah
G05	Cemas
G06	Konsentrasi menurun
G07	Jantung berdebar
G08	Sesak napas
G09	Turunnya tekanan darah.
G10	Sulit dan nyeri saat buat air kecil
G11	Urin keruh atau berdarah
G12	Nyeri perut bagian bawah
G13	Tekanan darah naik sampai 140/90mmHg
G14	Nyeri otot
G15	Kelelahan atau sakit kepala
G16	Jarang buang air kecil
G17	Berat badan naik mendadak
G18	Mau pingsan
G19	Pembengkakan pada kaki
G20	Pandangan kabur
G21	Menggigil

G22	Sakit perut bagian atas
G23	Demam
G24	Gelisah
G25	Pingsan
G26	Kulit ujung-ujung jari dan bibir membiru
G27	Merasa pusing
G28	Perasaan sedih / menangis tanpa alas an
G29	Sulit tidur (Imsomnia)
G30	Lemas
G31	Cepat marah
G32	Gampang tersinggung

Tabel 3. 3 Tabel Penyakit.

Kode	Nama Penyakit
P01	Pendarahan Postpartum
P02	Infeksi Postpartum
P03	Preeklampsia
P04	Embolik Paru
P05	Baby Blues

Tabel 3. 4 Daftar Hubungan Gejala dan Penyakit.

Gejala	Pendarahan Postpartum	Preeklampsia	Infeksi Postpartum	Embolik Paru	Baby Blues
G01	✓				
G02	✓				
G03	✓				
G04		✓			
G05					✓
G06					✓
G07	✓				
G08				✓	
G09	✓				
G10			✓		
G11			✓		

G12			v		
G13		v			
G14			v		
G15			v		
G16		v			
G17		v			
G18	v				v
G19		v			
G20		v			
G21			v		
G22		v			
G23			v	v	
G24				v	
G25				v	
G26				v	
G27	v			v	
G28					v
G29					v
G30	v				
G31					v
G32					v

Misalkan untuk penyakit emboli paru yaitu seorang pasien merasakan gejala yang dipilih adalah [G04] pusing dengan nilai pasiennya 0,4, [G26] kulit ujung-ujung jari dan bibir membiru, [G27] Sesak napas. Pada contoh kasus perlu diketahui bahwa pasien menginputkan gejala-gejala yang terdapat pada penyakit emboli paru. Sehingga pada perhitungan menggunakan metode *Certainty Factor* akan mendapatkan hasil diagnosa berdasarkan hasil persentase paling tinggi. Dibawah ini adalah gejala table berdasarkan nilai MB (nilai dari pakar) dan MD (nilai dari user).

Dan untuk penyakit infeksi postpartum yaitu seorang pasien merasakan gejala yang dipilih adalah [G11] urin keruh atau berdarah dengan nilai pasien 0,4, [G13] mengigil, [G14] nyeri otot. Pada contoh kasus perlu diketahui bahwa pasien

mengimputkan gejala-gejala yang terdapat penyakit infeksi postpartum. Sehingga pada perhitungan menggunakan metode *Certainty Factor* akan mendapatkan hasil diagnosa berdasarkan hasil persentase paling tinggi.

a. .Penyakit Emboli Paru

Tabel 3. 5 Emboli Paru

<b>Nama Penyakit</b>	<b>Kode Gejala</b>	<b>Nama Gejala</b>	<b>MB</b>	<b>MD</b>
<b>Preeklamsia</b>	G08	Sesak Napas	1,0	0,4
	G23	Demam	0,6	0,4
	G24	Gelisah	0,4	0,4
	G25	Pingsan	0,8	0,4
	G26	Kulit ujung-ujung jari dan bibir membiru	0,8	0,6
	G20	Merasa Pusing	0,6	0,4

b. Penyakit Infeksi Postpartum

Tabel 3. 6 Penyakit Infeksi Postpartum.

<b>Nama Penyakit</b>	<b>Kode Gejala</b>	<b>Nama Gejala</b>	<b>MB</b>	<b>MD</b>
<b>Infeksi Postpartum</b>	G10	Sulit dan nyeri saat buang air kecil	0,8	0,6
	G11	Urin keruh / berdarah	0,4	0,4
	G12	Nyeri perut bawah	0,4	0,4
	G23	Deman	0,6	0,6
	G13	Menggigil	0,6	0,4
	G14	Nyeri Otot	0,6	0,6
	G15	Kelelahan / Sakit Kepala yang keras	0,8	0,6

2. Eksperimen dan Cara Pengujian Model

a. Penyakit Emboli Paru

Karena adanya kaidah yang sama dan lebih dari satu gejala maka dihitung seperti perhitungan dibawah ini:

Yang pertama yaitu untuk menghitung gejala-gejala nilai dari pakar:

$$1. \ MB_1 = [Demam, 0,6] + [gelisah, 0,4] * (1-[MB \ gejala \ 1, 0,6])$$

$$= 0,6 + 0,4 * (1-0,6)$$

$$= 0,6 + 0,4 * (0,4)$$

$$= 0,6 + 0,16$$

$$= 0,56$$

$$2. \ MB_2 = [MB_1, 0,56] + [Merasa Pusing, 0,6 ] * (1- [MB_1, 0,56])$$

$$= 0,56 + 0,6 * (1-0,56)$$

$$= 0,56 + 0,6 * (0,44)$$

$$= 0,56 + 0,264$$

$$= 0,824$$

$$3. \ MB_3 = [MB_2, 0,824] + [Pingsan, 0,8] * (1- [MB_2, 0,824])$$

$$= 0,824 + 0,8 * (1-0,824)$$

$$= 0,824 + 0,8 * (0,176)$$

$$= 0,824 + 0,1408$$

$$= 0,9648$$

$$4. \ MB_4 = [MB_3, 0,9648 ] + [Kulit \ ujung-ujung \ jari \ dan \ bibir \ membiru , 0,8] * (1- [MB_3, 0,9648])$$

$$= 0,9648 + 0,8 * (1- 0,9648)$$

$$= 0,9648 + 0,8 * (0,0352)$$

$$= 0,9648 + 0,02816$$

$$= 0,99296$$

$$\begin{aligned}
5. \text{ MB}_5 &= [\text{MB}_4, 0,99296] + [\text{Sesak Napas}, 1,0] * (1 - [\text{MB}_4, 0,99296]) \\
&= 0,99296 + 1,0 + (1-0,99296) \\
&= 0,99296 + 1,0 + (0,00704) \\
&= 0,99296 + 0,00704 \\
&= 1
\end{aligned}$$

Yang kedua untuk menghitung dari gejala-gejala nilai dari user:

$$\begin{aligned}
6. \text{ MD}_1 &= [\text{Demam}, 0,4] + [\text{Gelisah}, 0,4] * (1 - [\text{MD gejala 1}, 0,4]) \\
&= 0,4 + 0,4 * (1-0,4) \\
&= 0,4 + 0,4 * (0,4) \\
&= 0,4 + 0,6 \\
&= 0,64 \\
7. \text{ MD}_2 &= [\text{MD}_1, 0,64] + [\text{Merasa Pusing}, 0,4] * (1 - [\text{MD}_1, 0,64]) \\
&= 0,64 + 0,4 * (1-0,64) \\
&= 0,64 + 0,4 * (0,36) \\
&= 0,7 + 0,144 \\
&= 0,748 \\
8. \text{ MD}_3 &= [\text{MD}_2, 0,748] + [\text{Pingsan}, 0,4] * (1 - [\text{MD}_2, 0,748]) \\
&= 0,748 + 0,4 * (1-0,748) \\
&= 0,748 + 0,4 * (0,216) \\
&= 0,748 + 0,0864 \\
&= 0,08704 \\
9. \text{ MD}_4 &= [\text{MD}_3, 0,08704] + [\text{Kulit ujung-ujung jari dan bibir membiru}, 0,6] * (1 - [\text{MD}_3, 0,08704]) \\
&= 0,08704 + 0,6 * (1-0,08704) \\
&= 0,08704 + 0,6 * (0,1296) \\
&= 0,08704 + 0,07776 \\
&= 0,94816 \\
10. \text{ MD}_5 &= [\text{MD}_4, 0,94816] + [\text{Sesak Napas}, 0,6] * (1 - [\text{MD}_4, ]) \\
&= 0,94816 + 0,6 * (1-0,94816) \\
&= 0,94816 + 0,6 * 0,05184 \\
&= 0,94816 + 0,031104
\end{aligned}$$

$$= 0,979264$$

$$\begin{aligned} \text{CF[h,e]} &= \text{MB[h,e]} - \text{MD[h,e]} \\ &= 1 - 0,979264 \\ &= 0,020736 \end{aligned}$$

Setelah dilakukan perhitungan dengan menggunakan metode *Certainty Factor*, maka di dapatkan nilai sebesar 0,020736.

b. Penyakit Infeksi Postpartum

1.  $\text{MB}_1 = [\text{Sulit dan nyeri saat buang air kecil}, 0,8] + [\text{Urin keruh dan berdarah}, 0,4] * (1-\text{MB}_1, 0,8)$   
 $= 0,8 + 0,4 * (1-0,8)$   
 $= 0,8 + 0,4 * 0,2$   
 $= 0,8 + 0,08$   
 $= 0,88$
2.  $\text{MB}_2 = [\text{MB}_1, 0,88] + [\text{Nyeri perut bawah}, 0,4] * (1-\text{MB}_1, 0,88)$   
 $= 0,88 + 0,4 * (1-0,88)$   
 $= 0,88 + 0,4 * (0,12)$   
 $= 0,88 + 0,048$   
 $= 0,928$
3.  $\text{MB}_3 = [\text{MB}_2, 0,928] + [\text{Demam}, 0,6] * (1-\text{MB}_1, 0,928)$   
 $= 0,928 + 0,6 * (1-0,928)$   
 $= 0,928 + 0,6 * (0,072)$   
 $= 0,928 + 0,0432$   
 $= 0,9712$
4.  $\text{MB}_4 = [\text{MB}_3, 0,9712] + [\text{Menggigil}, 0,6] * (1-\text{MB}_1, 0,9712)$   
 $= 0,9712 + 0,6 * (1-0,9712)$   
 $= 0,9712 + 0,6 * (0,0288)$   
 $= 0,9712 + 0,01728$   
 $= 0,98848$

5.  $MB_5 = [MB_4, 0,98848] + [Nyeri Otot, 0,6] * (1-[MB_1, 0,98848])$   
 $= 0,98848 + 0,6 * (1- 0,98848)$   
 $= 0,98848 + 0,6 * (0,01152)$   
 $= 0,992 + 0,006912$   
 $= 0,995392$
6.  $MB_6 = [MB_5, 0,995392] + [Kelelahan / sakit kepala yang keras, 0,8] * (1-[MB_1, 0,995392])$   
 $= 0,995392 + 0,8 * (1-0,995392)$   
 $= 0,995392 + 0,8 * (0,004608)$   
 $= 0,995392 + 0,0036864$   
 $= 0,9990784$
7.  $MD_1 = [Sulit dan nyeri saat buang air kecil, 06] + [Urin keruh dan berdarah, 0,4] * (1-[MD gejala 1, 0,6])$   
 $= 0,6 + 0,4 * (1-0,6)$   
 $= 0,6 + 0,4 * (0,4)$   
 $= 0,6 + 0,16$   
 $= 0,76$
8.  $MD_2 = [MD_1, 0,76] + [Nyeri perut bawah, 0,4] * (1-[MD_1 1, 0,76])$   
 $= 0,76 + 0,4 * (1-0,76)]$   
 $= 0,68 + 0,5 * (0,24)$   
 $= 0,68 + 0,096$   
 $= 0,856$
9.  $MD_3 = [MD_2, 0,84] + [Demam, 0,6] * (1-[MD_2, 0,84])$   
 $= 0,856 + 0,5 * (1-0,856)$   
 $= 0,84 + 0,6 * (0,144)$   
 $= 0,84 + 0,0864$   
 $= 0,9424$
10.  $MD_4 = [MD_3, 0,9424] + [Menggigil, 0,4] * (1-[MD_3, 0,9424])$   
 $= 0,9424 + 0,4 * (1-0,9424)$   
 $= 0,9424 + 0,4 * (0,0576)$   
 $= 0,9424 + 0,02304$   
 $= 0,96544$
11.  $MD_5 = [MD_4, 0,96544] + [Nyeri otot, 0,6] * (1-[MD_4 , 0,96544])$   
 $= 0,96544 + 0,6 * (1-0,96544)$   
 $= 0,96544 + 0,6 * (0,03456)$   
 $= 0,96544 + 0,020736$   
 $= 0,986176$

$$12. MD_6 = [MD_5, 0,986176] + [Kelelahan / sakit kepala yang keras, 0,6] * (1-[MD_5 , 0,986176])$$

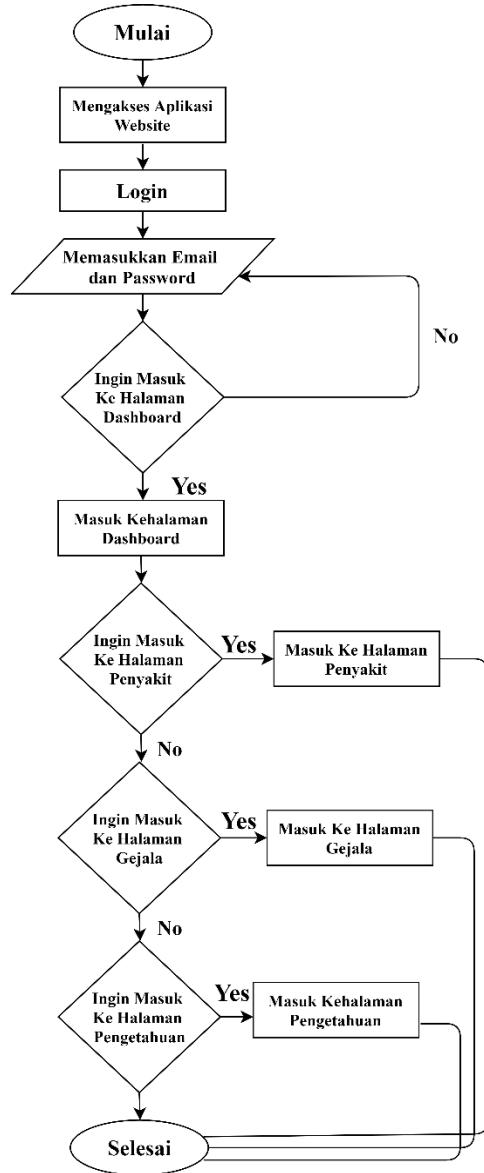
$$\begin{aligned} &= 0,986176 + 0,6 * (1-0,986176) \\ &= 0,986176 + 0,6 * (0,013824) \\ &= 0,986176 + 0,0082944 \\ &= 0,9944704 \end{aligned}$$

$$\begin{aligned} CF[h,e] &= MB[h,e] - MD[h,e] \\ &= 0,9990784 - 0,9944704 \\ &= 0,004608 \end{aligned}$$

Setelah dilakukan perhitungan dengan menggunakan metode *Certainty Factor*, maka didapatkan hasil nilai sebesar = 0,004608 dan diketahui bahwa pasien sedang menderita penyakit infeksi postpartum.

b. *Flowchart*

1. *Flowchart Admin*

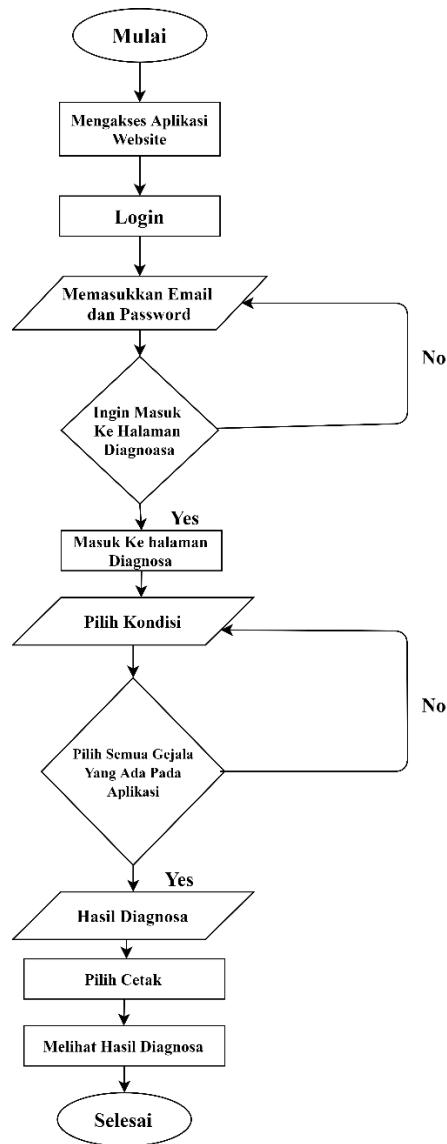


Gambar 3. 2 Flowchart Admin.

Pada *Flowchart* pengguna diatas alursistemnya yaitu mulai, proses mengakses aplikasi website, kemudian masuk ke proses halaman login, setelah melakukan login jika gagal maka kembali ke halaman login tetapi jika berhasil maka masuk pada proses halaman dashboard, setelah itu admin ingin mengelola data gejala, dan mengelola data penyakit,, maka apa bila bisa masuk ke proses halaman data gejala dan data penyakit, untuk mengelola data gejala dan data penyakit. Setelah masuk

pada halaman admin mengelola kriteria dan halaman mengelola gejala dan penyakit, serta proses selesai atau berakhir.

## 2. Flowchart User



Gambar 3. 3 Flowchart User.

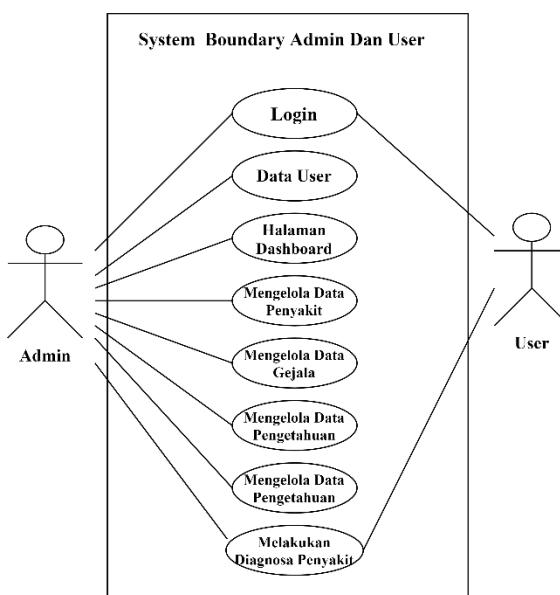
Pada *Flowchart User* diatas alur sistemnya yaitu mulai masuk pada proses akses aplikasi website, selanjutnya masuk ke halaman login dan kemudian setelah login berhasil masuk pada proses halaman diagnosa penyakit, selanjutnya masuk pada proses halaman diagnosa, setelah itu pilih kondisi, setelah melakukan pilih kondisi

jika masih ada kembali ke pilih kondisi jika tidak maka keluar hasil diagnosa, selanjutnya masuk pada proses cetak, keluar hasil cetak diagnosa. Sistem berakhir setelah proses melihat hasil diagnosa.

c. UML (*Unified Modeling Language*)

Dalam penelitian ini hanya menggunakan lima yaitu, *Flowchart*, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Classs Diagram*.

1. *Use Case Diagram*



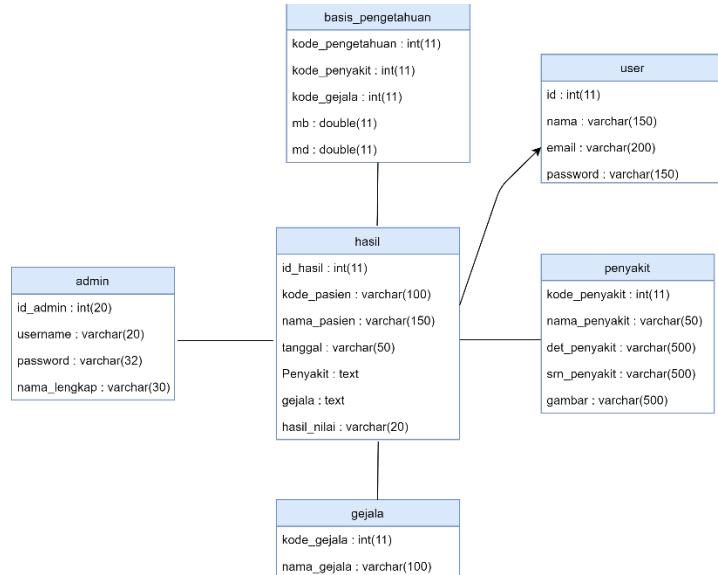
Gambar 3. 4 Use Case.

Tabel 3. 7 Deskripsi Use Case.

No	Use Case	Deskripsi
1.	Login	Login merupakan proses admin untuk masuk ke dalam aplikasi,yaitu untuk mengelola data penyakit, mengelola data gejala, dan mengelola data pengetahuan.
2.	Halaman beranda	Halaman beranda yaitu halaman awal dari aplikasi sitem pakar yang tampilannya ada beberapa tombol yaitu

		ada tombol 44iagnose, penyakit, gejala, dan pengetahuan.
3.	Mengelola data penyakit	Mengelola data penyakit merupakan proses generalisasi yang meliputi lima buah proses pengelolaan data penyakit yaitu memasukkan data penyakit, mengubah data penyakit, mencari penyakit, dan melihat penyakit.
4.	Mengelola data gejala	Mengelola data gejala merupakan proses generalisasi yang meliputi lima buah proses pengelolaan data penyakit yaitu memasukkan data gejala, mengubah data gejala, mencari gejala, dan melihat gejala.
5.	Mengelola data pengetahuan	Mengelola data pengetahuan merupakan proses generalisasi yang meliputi lima buah proses pengelolaan data pengetahuan yaitu memasukkan data pengetahuan,
6.	Melakukan diagnosa penyakit	Melakukan diagnosa penyakit merupakan proses penentuan untuk jenis penyakit
7.	Menampilkan hasil diagnosa penyakit	Menampilkan hasil diagnosa merupakan hasil dari diagnosa gejala yang dialami.

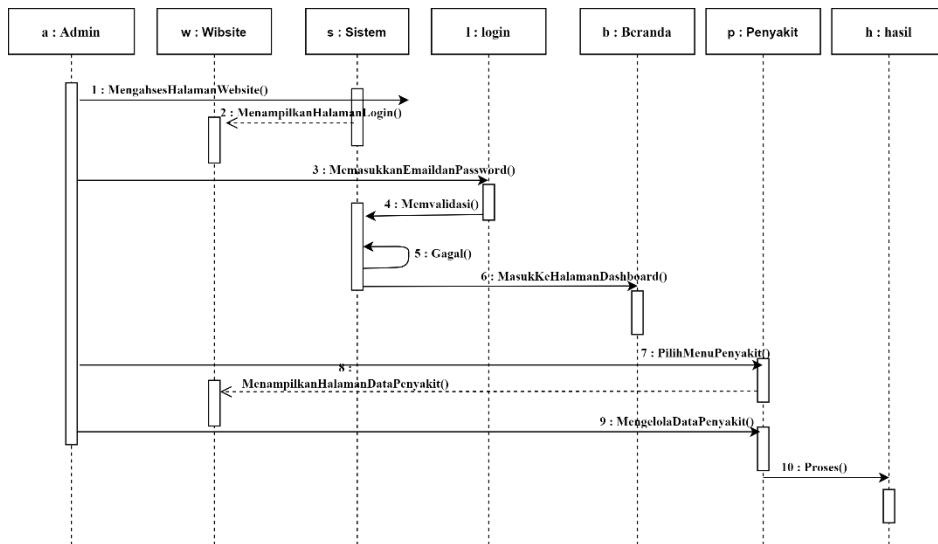
## 2. Class Diagram



Gambar 3. 5 Class Diagram.

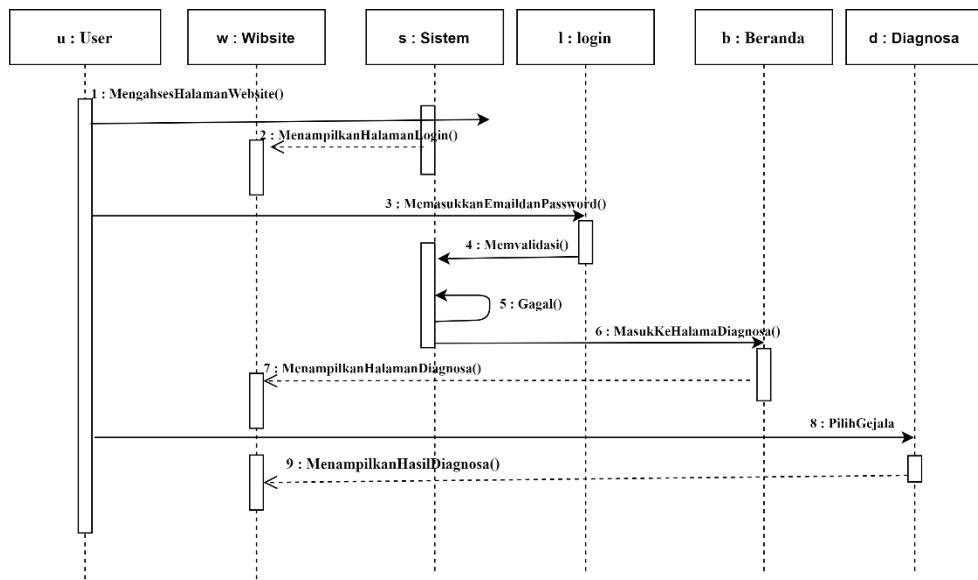
### 3. Sequence Diagram

#### a. Sequence Diagram Admin Mengelola Data Penyakit



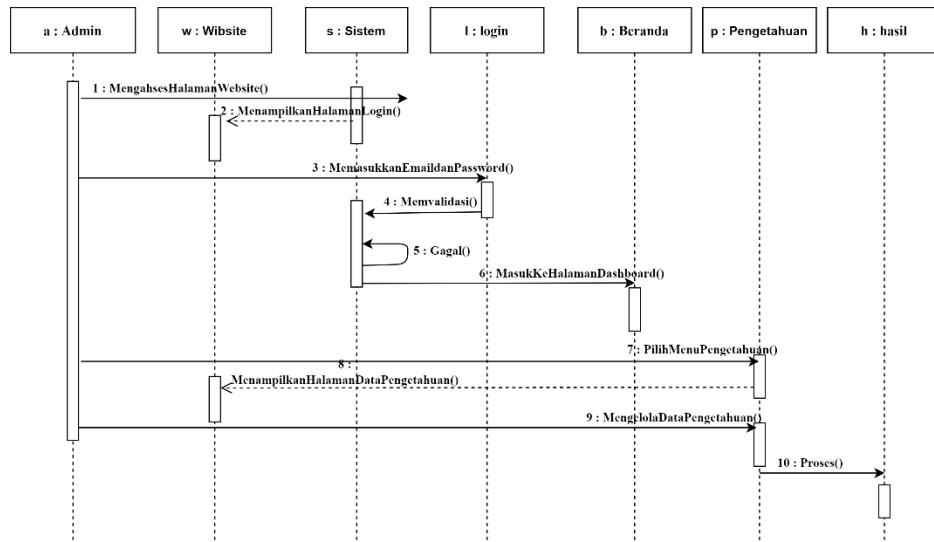
Gambar 3. 6 Sequence Diagram Admin Mengelola Data Penyakit.

#### b. Sequence Diagram Melakukan Diagnosa Penyakit



Gambar 3. 7 Sequence Diagram Melakukan Diagnosa Penyakit.

c. *Sequence Diagram Admin Mengelola Data Pengetahuan*

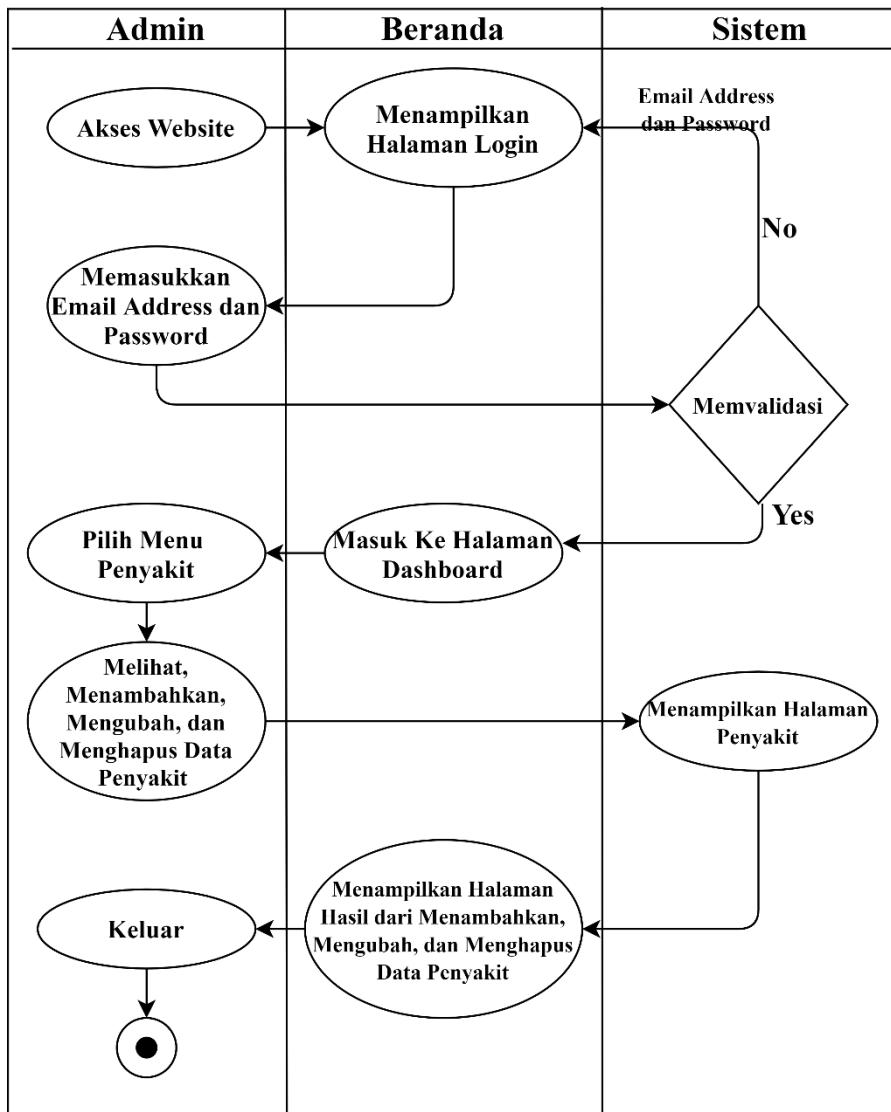


Gambar 3. 8 Sequence Diagram Admin Mengelola Data Pengetahuan.

4. *Activiy Diagram*

a. *Activiy Diagram Admin Mengelola Data Pengetahuan*

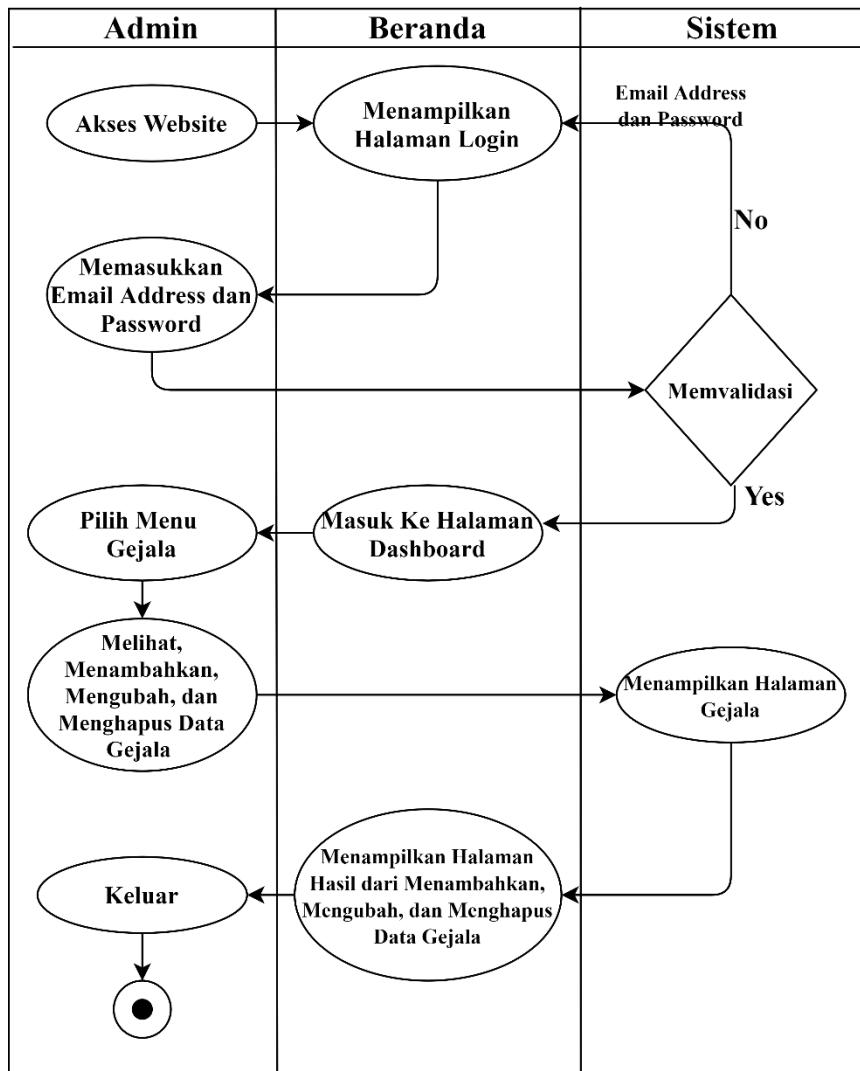
Aktivitas diagram admin mengelola data penyakit yaitu tugas admin yang mengelola data-data penyakit baik untuk melihat data penyaki, menambah data penyakit, mengubah data penyakit, dan bisa menghapus data penyakit. Adapun untuk aktivitas diagram admin mengelola data penyakit dapat dilihat dari gambar 3.9.



Gambar 3. 9 Diagram Aktivitas Mengelola Data Penyakit.

b. *Activity Diagram* Admin Mengelola Data Gejala

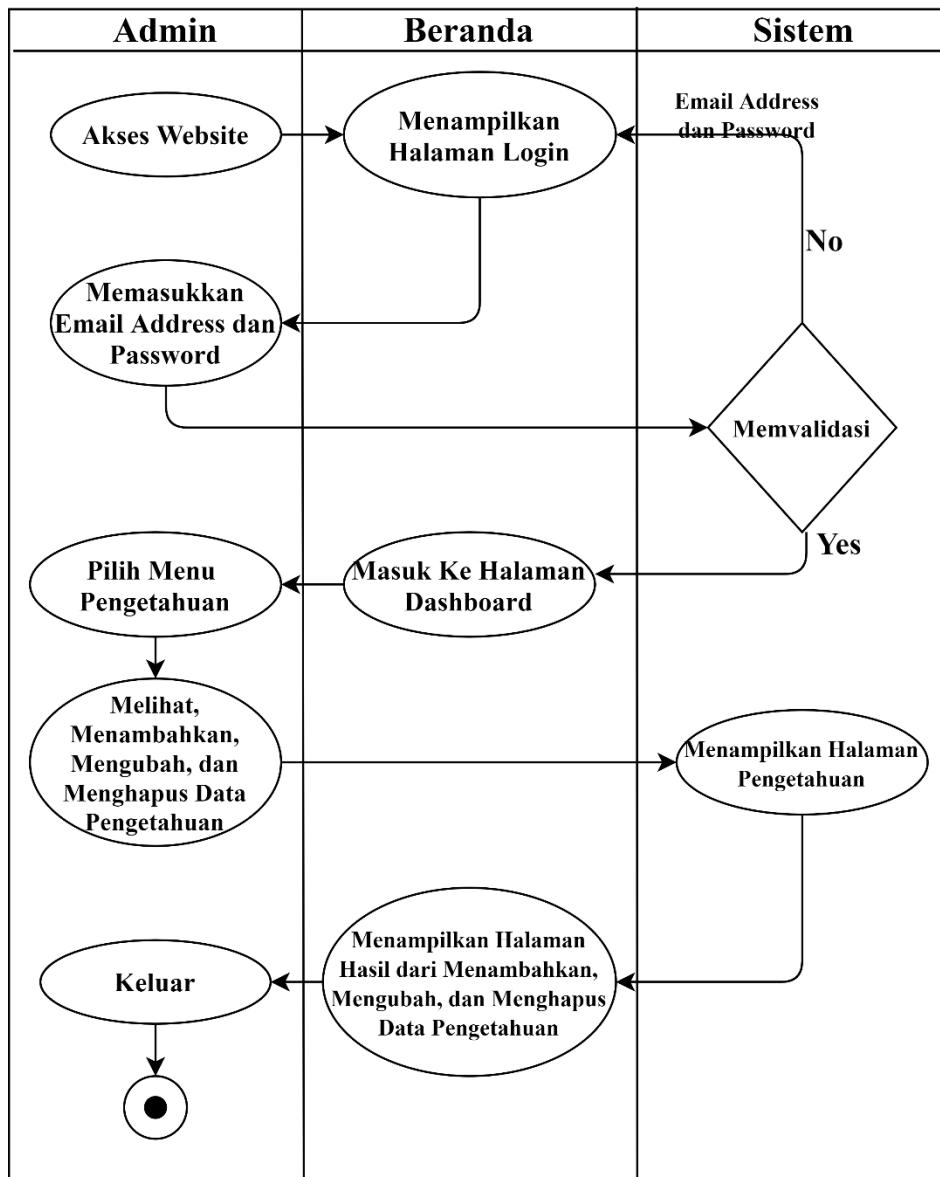
Aktivitas diagram admin mengelola data gejala yaitu tugas admin yang mengelola data-data gejala baik untuk melihat data gejala, menambah data gejala, mengubah data gejala, dan bisa menghapus data gejala. Adapun untuk aktivitas diagram admin mengelola data gejala dapat dilihat dari gambar 3.10.



Gambar 3. 10 Aktivitas Diagram Mengelola Data Gejala.

### c. *Activity Diagram Admin Mengelola Data Pengetahuan*

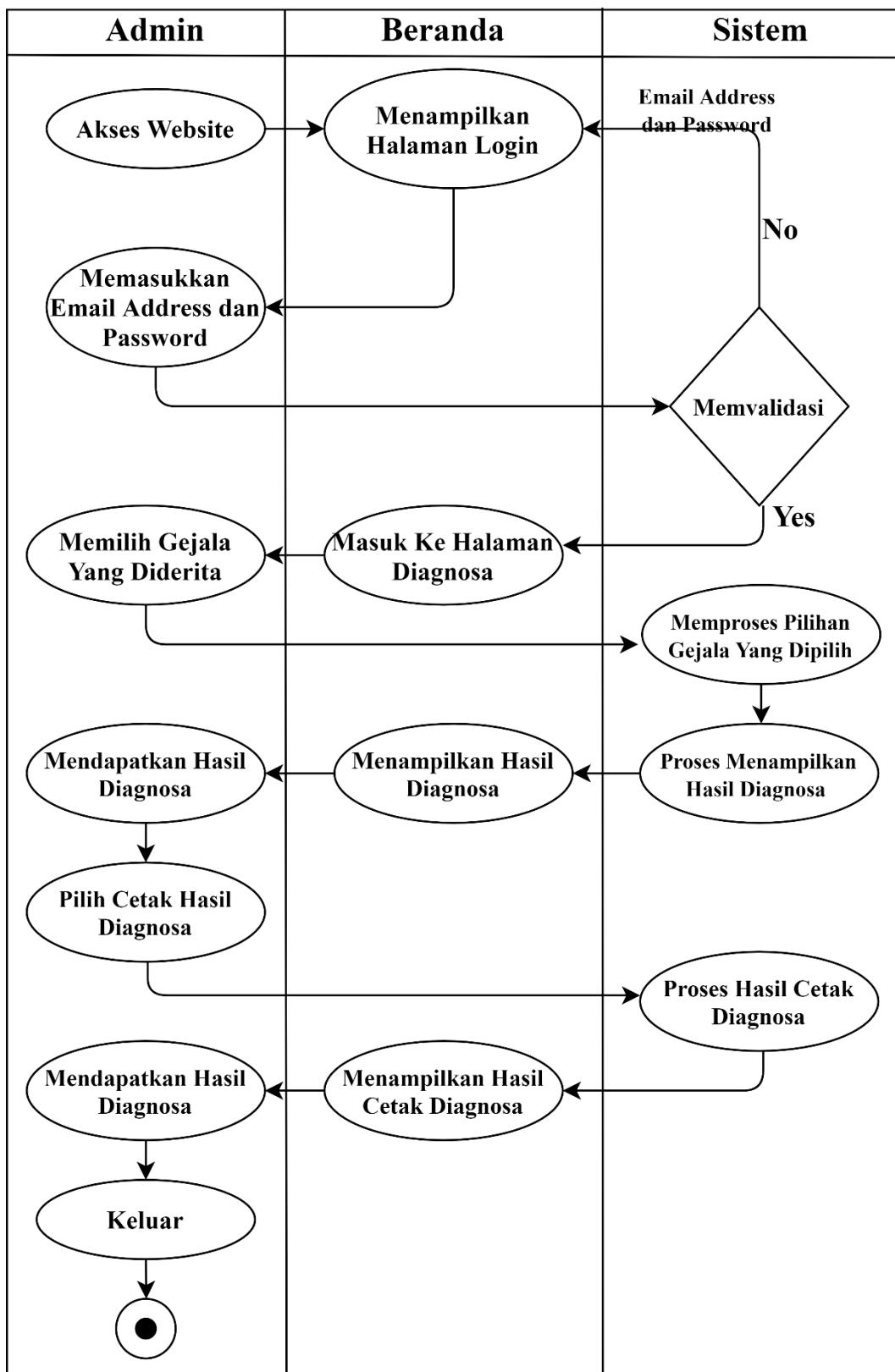
Aktivitas diagram admin mengelola data pengetahuan yaitu tugas admin yang mengelola data-data pengetahuan baik untuk melihat data pengetahuan, menambah data pengetahuan, mengubah data pengetahuan, dan bisa menghapus data pengetahuan. Adapun untuk aktivitas diagram admin mengelola data pengetahuan dapat dilihat dari gambar 3.11.



Gambar 3. 11 Aktivitas Diagram Mengelola Data Pengetahuan.

d. *Activity Diagram* Proses Diagnosa Penyakit

Aktivitas diagram proses diagnosa penyakit yaitu proses untuk mengetahui hasil diagnosa penyakit yang dialami ibu pasca melahirkan. Dengan langkah pertama yaitu memilih gejala yang diderita setelah itu sistem melakukan proses gejala yang telah dipilih yang nantinya akan mendapatkan hasil diagnosa, selanjutnya hasil diagnosa tadi dapat dilihat dan dicetak.



Gambar 3. 12 Aktivitas Diagram Proses Diagnosa Penyakit.

d. Perencanaan table basis data

1. Tabel User

Tabel 3. 8 Tabel User.

Nama	Tipe Data	Keterangan
Id	int(11)	Berisi dari kondisi
Name	varchar(150)	Berisi nama pasien
Email	varchar(200)	Berisi email
Password	varchar(150)	Berisi password
Gambar	varchar(150)	Berisi gambar

2. Tabel Admin

Tabel 3. 9 Tabel admin.

Nama	Tipe Data	Keterangan
Username	varchar(20)	Username
Password	varchar(32)	Password
nama_lengkap	varchar(30)	Nama lengkap

3. Tabel Penyakit

Tabel 3. 10 Tabel Penyakit.

Nama	Tipe Data	Keterangan
kode_penyakit	Int(11)	Kode penyakit
nama_penyakit	varchar(50)	Nama penyakit
det_penyakit	varchar(500)	Definisi penyakit
srn_penyakit	varchar(500)	Saran penyakit
Gambar	varchar(500)	Berisi gambar penyakit

4. Tabel Gejala

Tabel 3. 11 Tabel Gejala.

Nama	Tipe Data	Keterangan
kode_gejala	Int(11)	Kode gejala
nama_gejala	varchar(100)	Nama gejala

## 5. Tabel Basis Pengetahuan

Tabel 3. 12 Tabel Basis Pengetahuan.

Nama	Tipe Data	Keterangan
kode_pengetahuan	Int(11)	Kode pengetahuan
kode penyakit	int(11)	Kode penyakit
kode_gejala	int(11)	Kode gejala
Mb	double(11)	Nilai dari pakar
Md	double(11)	Nilai dari user

## 6. Tabel Kondisi

Tabel 3. 13 Tabel Kondisi.

Nama	Tipe Data	Keterangan
Id	int(11)	Id kondisi
Kondisi	varchar(64)	Pilihan kondisi yang rasa
Ket	varchar(256)	Keterangan

## 7. Tabel Hasil

Tabel 3. 14 Tabel Hasil.

Nama	Tipe Data	Keterangan
id_hasil	int(11)	Id hasil diagnosa
kode_pasien	varchar(100)	Kode pasien
nama_pasien	varchar(150)	Nama pasien
Penyakit	text(-)	Nama penyakit yang di derita
Gejala	text(-)	Nama gejala yang di derita
tgl	datetime	Berisi tanggal saat melakukan diagnosa
hasil_nilai	varchar(20)	Hasil nilai dari diagnosa

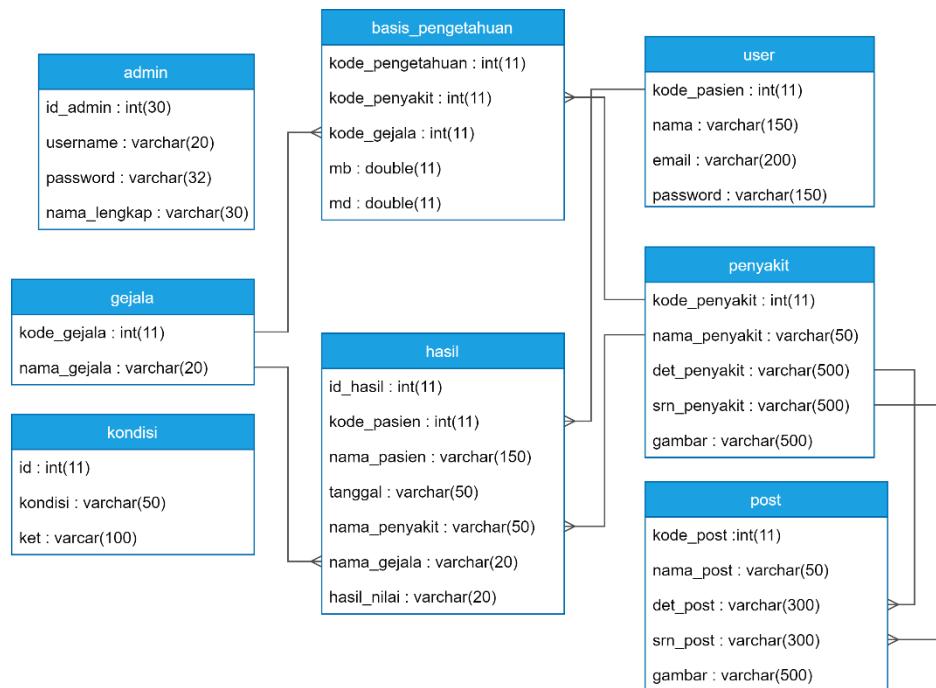
## 8. Tabel Post

Tabel 3. 15 Tabel Post.

Nama	Tipe Data	Keterangan
kode_keterangan	int(11)	Berisi kode keterangan

nama_post	varchar(50)	Berisi nama penyakit
def_post	varchar(15000)	Definisi keterangan penyakit
srn_post	varchar(15000)	Saran dari penyakit
Gambar	varchar (500)	Berisi gambar

e. Relasi antar table



Gambar 3. 13 Relasi Antar Tabel.

f. Desain interface

Desain interface merancang sebuah model pengembangan sistem yang akan dibangun.

1. Desain Halaman Login Admin

The diagram illustrates the layout of the Admin Login page. It features a central title 'Login Admin' at the top. Below the title are two input fields: the first for 'Username' and the second for 'Password'. At the bottom of the page is a single 'Login' button.

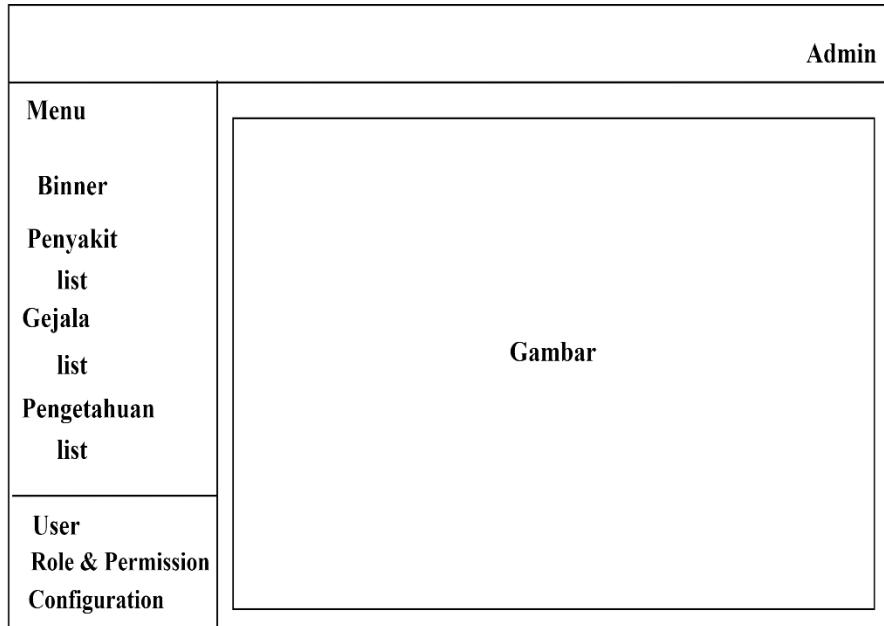
Gambar 3. 14 Desain Halaman Login Admin.

2. Desain Halaman Login User

The diagram illustrates the layout of the User Login page. It features a central title 'Login User' at the top. Below the title are two input fields: the first for 'Email Address' and the second for 'Password'. At the bottom of the page is a single 'Login' button.

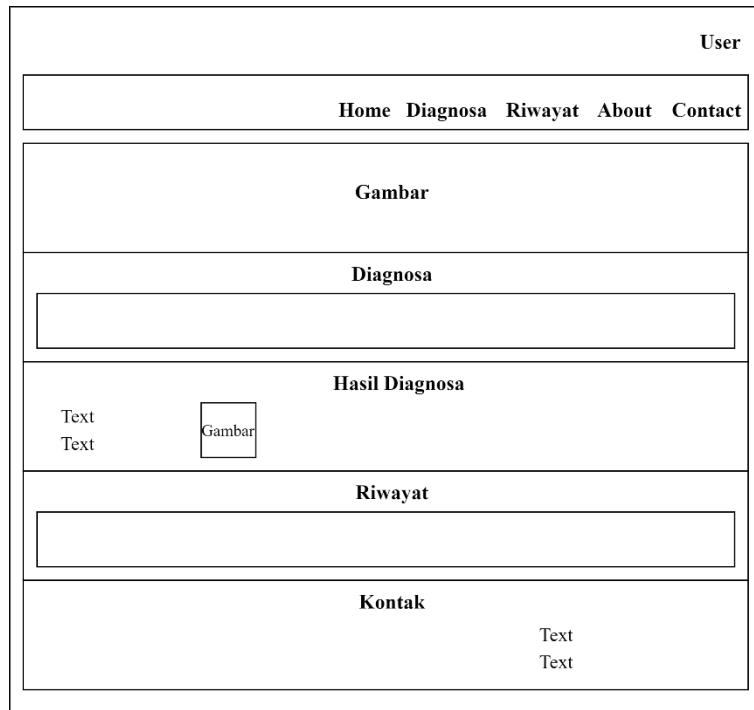
Gambar 3. 15 16 Desain Halaman Login User.

### 3. Desain Halaman Dashboard



Gambar 3. 16 Desain Halaman Dashboard.

### 4. Desain Halaman User



Gambar 3. 17 Desain Halaman User.

## 5. Desain Halaman Penyakit

Admin																		
<b>Menu</b>  <b>Binner</b> <b>Penyakit</b> list  <b>Gejala</b> list  <b>Pengetahuan</b> list	<b>Penyakit</b> <div style="float: right;"> <input type="button" value="Create Penyakit"/> </div> <div style="clear: both; margin-top: 10px;">         Search : <input type="text"/> </div> <div style="margin-top: 10px;"> <table border="1"> <thead> <tr> <th>Id</th> <th>Nama Penyakit</th> <th>Detail Penyakit</th> <th>Saran Penyakit</th> <th>Gambar</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td> <input type="button" value="Edit"/>  <input type="button" value="Delete"/> </td> </tr> </tbody> </table> </div>						Id	Nama Penyakit	Detail Penyakit	Saran Penyakit	Gambar	Action						<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Id	Nama Penyakit	Detail Penyakit	Saran Penyakit	Gambar	Action												
						<input type="button" value="Edit"/> <input type="button" value="Delete"/>												

Gambar 3. 18 Desain Halaman Penyakit.

## 6. Desain Halaman Gejala

Admin												
<b>Menu</b>  <b>Binner</b> <b>Penyakit</b> list  <b>Gejala</b> list  <b>Pengetahuan</b> list	<b>Gejala</b> <div style="float: right;"> <input type="button" value="Create Gejala"/> </div> <div style="clear: both; margin-top: 10px;">         Search : <input type="text"/> </div> <div style="margin-top: 10px;"> <table border="1"> <thead> <tr> <th>Id</th> <th>Nama Penyakit</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> </td> </tr> </tbody> </table> </div>						Id	Nama Penyakit	Action			<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Id	Nama Penyakit	Action									
			<input type="button" value="Edit"/> <input type="button" value="Delete"/>									

Gambar 3. 19 Desain Halaman Gejala.

## 7. Desain Halaman Pengetahuan

Admin					
<b>Menu</b>	<b>Pengetahuan</b>				
Binner	<b>Create Pengetahuan</b>				
Penyakit	Search : <input type="text"/>				
list	<b>Id</b>	<b>Penyakit</b>	<b>Gejala</b>	<b>MB</b>	<b>MD</b>
Gejala					<b>Action</b>
list					<b>Edit</b> <b>Delede</b>
Pengetahuan					<b>Edit</b> <b>Delede</b>
list					

Gambar 3. 20 Desain Halaman Pengetahuan.

## 8. Desain Halaman Diagnosa

User											
<b>Text</b>											
<input type="text"/> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Gejala</th> <th>Pilih Kondisi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td>           Pilih Jika Sesuai            Pasti ya            Hampir pasti            Kemungkinan besar ya            Mungkin ya            Tidak tau            Mungkin tidak  <b>Hasil</b> </td> </tr> </tbody> </table>				No	Kode	Gejala	Pilih Kondisi				Pilih Jika Sesuai Pasti ya Hampir pasti Kemungkinan besar ya Mungkin ya Tidak tau Mungkin tidak <b>Hasil</b>
No	Kode	Gejala	Pilih Kondisi								
			Pilih Jika Sesuai Pasti ya Hampir pasti Kemungkinan besar ya Mungkin ya Tidak tau Mungkin tidak <b>Hasil</b>								

Gambar 3. 21 Desain Halaman Diagnosa.

9. Desain Halaman Hasil Diagnosa

<b>Hasil Diagnosa</b>	
<b>Jenis Penyakit Yang Diderita</b> _____	<b>Gambar</b> _____
<b>Detail</b> _____	
<b>Saran</b> _____	

Gambar 3. 22 Desain Halaman Hasil Diagnosa.

10. Desain Halaman Hasil Cetak

<b>Riwayat Pasien</b>				
No	Kode Pasien	Nama Pasien	Tanggal	Tanggal Diagnosa

Gambar 3. 23 Desain Halaman Hasil Cetak.

### 3.2 Instrumen Penelitian

Dalam sebuah pengumpulan data-data penyakit pasca melahirkan di rumah sakit Pembalih Batung Amuntai dilakukan dengan berbagai metode-metode penelitian seperti wawancara dan observasi. Wawancara dilakukan untuk mengetahui atau untuk mengumpulkan data-data penyakit pasca

melahirkan. Dan observasi yaitu sebagai alat bantu dalam pengumpulan data-data penyakit pasca melahirkan. Adapun untuk intrumen wawancara dan observasi seperti dibawah ini:

a. Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) yang digunakan antara lain:

1. Prosessor : Intel Core i3-5005U CPU @ 2.00Hz 2.00Hz
2. Intelled : 4,00 GB (3,68 GB usable)
3. System Type : 64 bit Operating Sytem, x64-based processor
4. Harddisk : 500 GB
5. Menggunakan *Mouse*, *Keyboard*, *Monitor* sebagai perantara antar muka.

b. Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) yang digunakan antara lain:

1. Sistem Operasi : Windows 10 pro 64-bit
2. Visual Studio Code : Teks editor
3. PHP : Bahasa pemograman yang digunakan
4. Laragon : Server
5. Google Chrome : Web browser

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak dari sistem yang akan dibangun secara lebih lengkap dijelaskan di bawah ini:

Perangkat Lunak:

- Sistem Operasi : Microsoft Windows 10 pro 64-bit
- Software : Laragon Full 4.0.16, Microsoft Visual Studio Code, PHP 7.2.11
- Basis Data : MySQL 5.7.24.

##### 4.1.1 Implementasi Basis Data

Pembuatan *database* dilakukan dengan menggunakan aplikasi *DBMS MySQL*. Implementasi *database* dalam Bahasa *SQL* adalah sebagai berikut:

a. Tabel Admin

Implementasi dari rancangan basis admin table data dapat dilihat pada gambar 4.1

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	username	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
2	password	varchar(32)	latin1_swedish_ci	Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
3	nama_lengkap	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya

Gambar 4. 1 Tabel Admin.

b. Tabel Basis Pengetahuan

Implementasi dari rancangan basis data table basis pengetahuan dapat dilihat pada gambar 4.2

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	kode_pengetahuan	int(11)		Tidak	Tidak ada		AUTO_INCREMENT		Ubah  Hapus ▾ Lainnya
2	kode_penyakit	int(11)		Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
3	kode_gejala	int(11)		Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
4	mb	double(11,1)		Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
5	md	double(11,1)		Tidak	Tidak ada				Ubah  Hapus ▾ Lainnya
6	created_at	timestamp		Ya	NULL				Ubah  Hapus ▾ Lainnya
7	updated_at	timestamp		Ya	NULL				Ubah  Hapus ▾ Lainnya

Gambar 4. 2 Basis Pengetahuan.

### c. Tabel Gejala

Implementasi dari rancangan basis data table gejala dapat dilihat pada gambar 4.3

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	kode_gejala	int(11)		Tidak	Tidak ada		AUTO_INCREMENT		Ubah Hapus Lainnya
2	nama_gejala	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	created_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
4	updated_at	timestamp		Ya	NULL				Ubah Hapus Lainnya

Gambar 4. 3 Tabel Gejala.

### d. Tabel Penyakit

Implementasi dari rancangan basis data table penyakit dapat dilihat pada gambar 4.4

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	kode_penyakit	int(11)		Tidak	Tidak ada		AUTO_INCREMENT		Ubah Hapus Lainnya
2	nama_penyakit	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	det_penyakit	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	srm_penyakit	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
5	gambar	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
6	created_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
7	updated_at	timestamp		Ya	NULL				Ubah Hapus Lainnya

Gambar 4. 4 Tabel Penyakit.

### e. Tabel Kondisi

Implementasi dari rancangan basis data table kondisi dapat dilihat pada gambar 4.5

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)		Tidak	Tidak ada		AUTO_INCREMENT		Ubah Hapus Lainnya
2	kondisi	varchar(64)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	ket	varchar(256)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Gambar 4. 5 Tabel Kondisi.

### f. Tabel post

Implementasi dari rancangan basis data table post dapat dilihat pada gambar 4.6

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	kode_post	int(11)		Tidak	Tidak ada		AUTO_INCREMENT		Ubah Hapus Lainnya
2	nama_post	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	det_post	varchar(15000)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	srm_post	varchar(15000)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
5	gambar	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
6	created_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
7	updated_at	timestamp		Ya	NULL				Ubah Hapus Lainnya

Gambar 4. 6 Tabel Post.

### g. Tabel User

Implementasi dari rancangan basis data table user dapat dilihat pada gambar 4.7

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	<input type="checkbox"/> id	int(10)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah
2	<input type="checkbox"/> name	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada			Ubah
3	<input type="checkbox"/> email	varchar(200)	latin1_swedish_ci		Tidak	Tidak ada			Ubah
4	<input type="checkbox"/> password	varchar(150)	latin1_swedish_ci		Ya	NULL			Ubah
5	<input type="checkbox"/> remember_token	text		latin1_swedish_ci	Ya				Ubah
6	<input type="checkbox"/> image	varchar(150)	latin1_swedish_ci		Ya	NULL			Ubah
7	<input type="checkbox"/> created_at	timestamp			on update CURRENT_TIMESTAMP	Tidak	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Ubah
8	<input type="checkbox"/> updated_at	timestamp				Tidak	0000-00-00 00:00:00		Ubah

Gambar 4. 7 Tabel User.

#### 4.1.2 Implementasi Interface

##### a. Pembuatan Program Halaman Login Admin

Halaman login admin adalah halaman yang pertama kali muncul sebelum masuk ke halaman admin. Pada halaman login admin ini akan menampilkan form login yang berisi email address dan password. Berikut potongan *Source code* halaman login admin :

```
<form method="POST" action="{{ route('login') }}>
@csrf
<div class="form-group">
    <label>Email address</label>
    <input id="email" type="email" class="form-control"
        @error('email') is-invalid @enderror name="email"
        value="{{ old('email') }}" autocomplete="email" autofocus>
        @error('email')
            <span class="invalid-feedback" role="alert">
                <strong>{{ $message }}</strong>
            </span>
        @enderror
    </div>
    <div class="form-group">
        <label>Password</label>
        <input id="password" type="password" class="form-control"
            @error('password') is-invalid @enderror name="password"
            autocomplete="current-password">
            @error('password')
                <span class="invalid-feedback" role="alert">
                    <strong>{{ $message }}</strong>
                </span>
            @enderror
        </div>
        <div class="checkbox">
            </div>
            <div class="form-group">
            </div>
            <button type="submit" class="btn btn-success btn-flat m-b-30
m-t-30">Sign in</button>
```

```
</form>
```

### 1. Pengujian White Box Halaman Login Admin

Pengujian *white Box* dilakukan dengan melakukan pengujian program (*coding*). Hasil *compile* halaman login admin berupa tampilan halaman login yang terdiri dari form email dan password. Halaman login admin terlihat pada Gambar 4.8.

Gambar 4. 8 Tampilan Halaman Login Admin.

Pada halaman login admin yaitu jika admin ingin masuk ke halaman utama admin untuk melakukan diagnosa penyakit pasca melahirkan, dengan mengisikan email dan password terlebih dahulu.

### 2. Pengujian Black Box Halaman Login Admin

Pengujian *black box* halaman login admin dilakukan dengan mengisikan email address dan password admin. Pengujian dan hasil uji *black box* halaman login admin pada Tabel 4.1.

Tabel 4. 1 Pengujian Black Box Menu Login Admin.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Form Email Address	Mengisikan email address dengan format yang benar	Sukses

	Mengisikan email address dengan format yang salah	Gagal
form Password	Mengisikan password harus huruf dan angka	Sukses
	Mengisikan password selain huruf dan angka	Gagal
Tombol Sign In	Masuk ke halaman dashboard	Sukses

### b. Pembuatan Program Halaman Login User

Halaman login user adalah halaman yang pertama kali muncul sebelum masuk ke halaman user. Pada halaman login user ini akan menampilkan form login yang berisi email dan password. Berikut potongan *Source code* halaman login user :

```
<form id="form_validation" method="POST" action="{{
route('users.store') }}"
@else
<form id="form_validation" method="POST" action="{{
route('users.update', $user->id) }}"
<input type="hidden" name="_method" value="PUT">
@endif
{{ csrf_field() }}
<div class="form-group form-float">
<div class="form-line">
<label class="form-label">Name User</label>
<input type="text" class="form-control" name="name"
maxlength="30" minlength="3" value="{{ $user->name }}"
required>
</div>
<div class="help-info">Min. 3 character, Max. 30
character</div>
</div>
<div class="form-group form-float">
<div class="form-line">
<label class="form-label">Email</label>
<input type="text" class="form-control" name="email"
maxlength="30" minlength="3" value="{{ $user->email }}"
required>
</div>
<div class="help-info"></div>
</div>
<div class="form-group form-float">
<div class="form-line">
<input type="password" class="form-control"
name="password" placeholder="password">
</div>
```

### 1. Pengujian White Box Halaman Login User

Pengujian halaman login user dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman berupa tampilan halaman user yang terdiri dari form login yang berisi email dan password. Halaman login user terlihat pada Gambar 4.9.

The image shows a user login interface. At the top center, it says "SELAMAT DATANG". Below that is a label "EMAIL ADDRESS" next to a text input field. Below that is a label "PASSWORD" next to another text input field. At the bottom is a large green rectangular button with the text "SIGN IN" in white.

Gambar 4. 9 Tampilan Halaman Login User.

Pada halaman login *user* yaitu jika *user* ingin masuk ke halaman utama *user* untuk melakukan diagnosa penyakit pasca melahirkan, dengan mengisikan email dan password terlebih dahulu.

### 2. Pengujian Black Box Halaman Login User

Pengujian *black box* halaman login *user* dilakukan dengan mengisikan email address dan password *user*. Pengujian dan hasil uji *block box* halaman login admin pada Tabel 4.2.

Tabel 4. 2 Pengujian Black Box Menu Halaman Login User.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Form Email Address	Mengisikan email address dengan format yang benar	Sukses
	Mengisikan email address dengan format yang salah	Gagal

form Password	Mengisikan password harus huruf dan angka	Sukses
	Mengisikan password selain huruf dan angka	Gagal
Tombol Sign In	Masuk ke halaman diagnosa	Sukses

### c. Pembuatan Program Halaman Dasboard

Halaman dashboard adalah halaman setelah admin melakukan login lalu masuk ke halaman dashboard. Halaman dashboard ini menampilkan menu-menu yang ada pada sistem yaitu menu daftar penyakit, daftar gejala, daftar pengetahuan, dan menampilkan daftar user dan admin. Berikut potongan *Source code* halaman Dasboard :

```
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false"> <i class="menu-icon fa fa-
bug"></i>Banner</a>
<ul class="sub-menu children dropdown-menu">
    <li><i class="fa fa-th"></i><a href="#"><{
        route('banners.index') }>list</a></li>
</ul>
</li>
<li class="menu-item-has-children dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false"> <i class="menu-
icon fa fa-bug"></i>Penyakit</a>
    <ul class="sub-menu children dropdown-menu">
        <li><i class="fa fa-th"></i><a href="#"><{
            route('penyakit.index') }>list</a></li>
    </ul>
    </li>
    <li class="menu-item-has-children dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false"> <i class="menu-icon fa fa-pencil"></i>Gejala</a>
        <ul class="sub-menu children dropdown-menu">
            <li><i class="fa fa-th"></i><a href="#"><{
                route('gejala.index') }>list</a></li>
        </ul>
        </li>
        <li class="menu-item-has-children dropdown">
            <a href="#" class="dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false"> <i class="menu-icon fa fa-
flask"></i>Pengetahuan</a>
            <ul class="sub-menu children dropdown-menu">
                <li><i class="fa fa-th"></i><a href="#"><{
                    route('pengetahuan.index') }>list</a></li>
            </ul>
            </li>
        </ul>
    </li>
</ul>
```

```

<li class="menu-item-has-children dropdown">
    <a href="#" class="dropdown-toggle" data-
        toggle="dropdown" aria-haspopup="true" aria-
        expanded="false"> <i class="menu-icon fa fa-file-
        text"></i>Keterangan</a>
    <ul class="sub-menu children dropdown-menu">
        <li><i class="menu-icon fa fa-th"></i><a href="{{
            route('keterangan.index') }}">list</a></li>
    </ul>
</li>

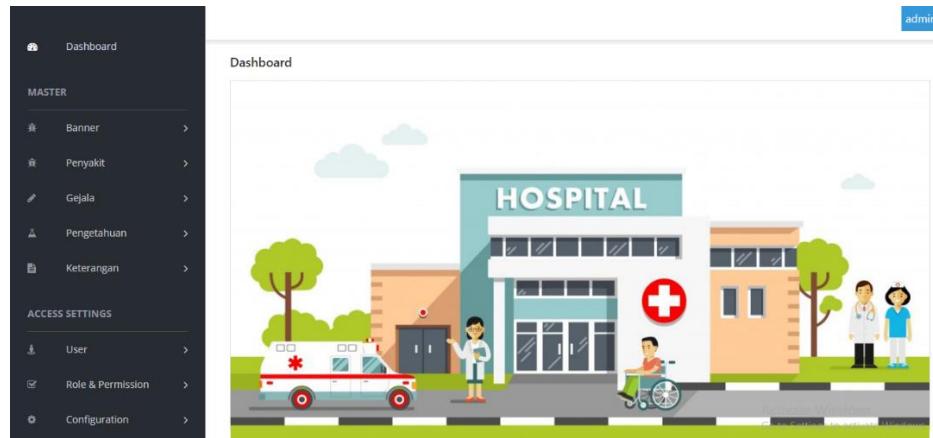
<h3 class="menu-title">Access Settings</h3><!-- /.menu-title -->
<li class="menu-item-has-children dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false"> <i
        class="menu-icon fa fa-street-view"></i>User</a>
    <ul class="sub-menu children dropdown-menu">
        <li><i class="menu-icon fa fa-th"></i><a href="{{
            route('users.index') }}">list</a></li>
    </ul>
</li>
<li class="menu-item-has-children dropdown">
    <a href="#" class="dropdown-toggle" data-
        toggle="dropdown" aria-haspopup="true" aria-
        expanded="false"> <i class="menu-icon fa fa-check-
        square-o"></i>Role & Permission</a>
    <ul class="sub-menu children dropdown-menu">
        <li><i class="menu-icon fa fa-shield"></i><a href="{{
            route('roles.index') }}">Role</a></li>
        <li><i class="menu-icon fa fa-share-alt"></i><a href="{{
            route('permissions.index') }}">Permission</a></li>
        <!-- <li><i class="menu-icon fa fa-diamond"></i><a
            href="{{ route('permissions.roles') }}">Permission
            role</a></li> -->
    </ul>
</li>
<li class="menu-item-has-children dropdown">
    <a href="{{ route('configurations.index') }}"> <i
        class="menu-icon fa fa-gear"></i>Configuration</a>
</li>

```

## 1. Pengujian White Box Halaman Dashboard

Pengujian *white box* dilakukan dengan melakukan pengujian lode program (*coding*). Hasil *compile* halaman dashboard berupa tampilan halaman yang terdiri dari menu banner, penyakit, gejala, pengetahuan, keterangan, *user*, *role & permission*, dan *configuration*.

Pengujian Halaman *user* tampilan menu *User* pada Gambar 4. 10.



Gambar 4. 10 Tampilan Halaman Dashboard.

## 2. Pengujian Black Box Halaman Dashboard

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman dashboard. Pengujian dan hasil uji *black box* halaman dashboard pada Tabel 4.3.

Tabel 4. 3 Pengujian Black Box Menu Halaman Dashboard.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik menu Banner	Muncul tampilan banner	Sukses
Klik menu Penyakit	Muncul tampilan daftar penyakit	Sukses
Klik menu Gejala	Muncul tampilan daftar Gejala	Sukses
Klik menu Pengetahuan	Muncul tampilan daftar Pengetahuan	Sukses
Klik menu User	Muncul tampilan daftar user	Sukses
Klik menu Role & permission	Muncul tampilan Role & permission	Sukses
Klik menu Configuration	Muncul tampilan Configuration	Sukses
Klik tombol Admin	Kembali pada halaman login	Sukses

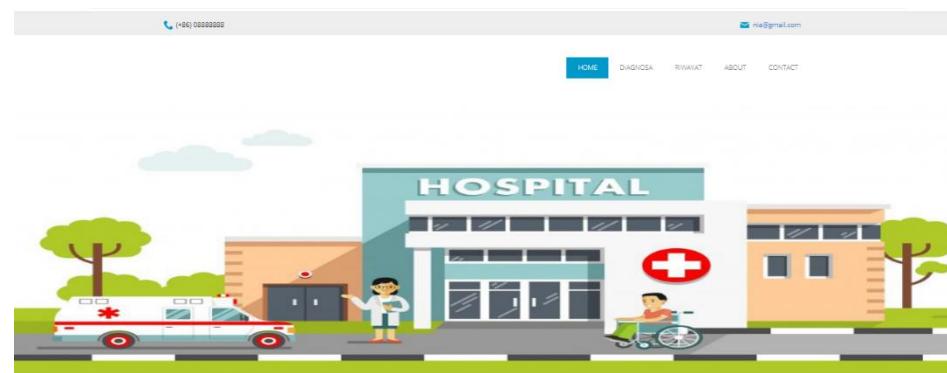
#### d. Pembuatan Program Halaman User

Halaman User adalah halaman setelah user login lalu masuk ke halaman user, pada halaman user ini akan menampilkan menu *home*, diagnosa, riwayat, *about*, dan *contact*. Berikut potongan *Source code* halaman user :

```
<div class="navbar-collapse collapse" id="templatemo-nav-bar">
    <ul class="nav navbar-nav navbar-right" style="margin-top:40px;">
        <li> <a href="#" class="js-scroll-to">HOME</a> </li>
        <li> <a href="#" class="js-scroll-to">DIAGNOSA</a> </li>
        <li> <a href="#" class="js-scroll-to">RIWAYAT</a> </li>
        </li>
        <li> <a href="#" class="js-scroll-to">CONTACT</a> </li>
        <li> <a class="dropdown-item" href="{{ route('logout') }}" onclick="event.preventDefault(); document.getElementById('logout-form').submit(); "> ('Logout') </a>
    </ul>
    <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
        @csrf
    </form>
```

#### 1. Pengujian *White Box* Halaman User

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman *User* yang terdiri dari menu *home*, diagnosa, riwayat, *about*, dan *contact*. Halaman utama terlihat pada Gambar 4.11.



Gambar 4. 11 Tampilan Halaman User.

## 2. Pengujian *Black Box* Halaman User

Pengujian *black box* dilakukan dengan mengujia setiap aktivitas yang terdapat pada halaman *user*. Pengujian dan hasil uji *black box* halaman user pada Tabel 4.4.

Tabel 4. 4 Pengujian Black Box Menu Halaman User.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik menu Home	Muncuk tampilan gambar	Sukses
Klik Menu Diagnosa	Muncul tampilan data Gejala	Sukses
Klik menu Riwayat	Muncuk tampilan riwayat hasil diagnosa pasien	Sukses
Klik menu Contact	Muncul tampilan Contact	Sukses

## e. Pembuatan Program Halaman Penyakit

Halaman penyakit adalah halaman yang ada pada admin setelah admin melakukan halaman login lalu masuk ke halaman dashboard, setelah itu klik menu penyakit. Halaman penyakit ini menampilkan semua data penyakit yang ada. pada halaman penyakit ini menampilkan menu-menu yang ada pada sistem yaitu menu *create* penyakit, *search*, *edit*, *delete*, *next*, dan *previous*. Berikut potongan *source code* halaman penyakit :

```
<div class="content">
    <div class="grid-system">
        <div class="row-center">
            <div class="card">
                @if (session('success'))
                    <div class="alert alert-success alert-dismissible">
                        <a class="close" data-dismiss="alert" href="#">X</a>
                        <h4 class="alert-heading">Sukses!</h4>
                        {{ session('success') }}
                    </div>
                @endif
                @if (session('edit'))
                    <div class="alert alert-success alert-dismissible">
                        <a class="close" data-dismiss="alert" href="#">X</a>
                        <h4 class="alert-heading">Sukses!</h4>
                        {{ session('edit') }}
                    </div>
                @endif
            <div class="container">
```

```
<div class="header">
    <h3>Penyakit</h3>
    </div>
    <div class="col-md-12">
        <div class="btn-group" role="group" style="padding-right: 50px; box-shadow: none !important; float: right;">
            <a class="btn btn-primary waves-effect" href="{{ route('penyakit.create') }}> Create Penyakit</a>
        </div>

        </div><br><br>
        <div class="body table-responsive">
            <table class="table table-bordered table-striped" id="penyakit">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>nama</th>
                        <th>detail</th>
                        <th>Saran</th>
                        <th>Gambar</th>
                        <th>Action</th>
                    </tr>
                </thead>
                </table>
            </div>
        </div>
    </div>
</div>
```

### 1. Pengujian White Box Halaman Penyakit

Pengujian *white box* dilakukan dengan melakukan pengujian kode program *coding*). Hasil *compile* halaman penyakit berupa tampilan halaman penyakit yang terdiri dari menu *create* penyakit, *Search*, *edit*, *delate*, *next*, dan *previous*. Halaman penyakit terlihat pada Gambar 4.12.

Penyakit					
				Create Penyakit	
ID	nama	detail	Saran	Gambar	Action
1	Pendarahan Postpartum	Pendarahan postpartum adalah pendarahan hebat yang terjadi pasca melahirkan. Penyebab paling umum dari pendarahan postpartum ini yaitu, terbukanya pembuluh darah pada rahim.			<a href="#">Edit</a> <a href="#">Delete</a>
2	Infeksi Postpartum	Infeksi postpartum yaitu terjadi pada saat pasca melahirkan atau usai persalinan, ada kemungkinan terjadi robeakan pada jalan lahir atau perineum, luka terbuka pada rahim di tempat menempelnya plasenta, atau luka pada operasi setelah oprasi ceser.			<a href="#">Edit</a> <a href="#">Delete</a>
3	Preeklamsia	Preeklamsia adalah komplikasi kehamilan serius yang ditandai dengan tekanan darah tinggi yang tidak terkontrol pada ibu hamil.			<a href="#">Edit</a> <a href="#">Delete</a>
4	Embolji Paru	Embolji paru adalah konsisi yang cukup berbahaya yang mengakibatkan oksigen terhambat, sehingga oksigen tidak mengalir pada tubuh dan merusak sistem tubuh.			<a href="#">Edit</a> <a href="#">Delete</a>
5	Baby Blues	Baby blues adalah suatu gangguan psikologis sementara yang ditandai dengan memuncaknya emosi pada minggu pertama pasca melahirkan. Baby blues adalah sebuah perasaan sedih tanpa dasar terjadi setelah melahirkan.			<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Gambar 4. 12 Tampilan Halaman Penyakit.

## 2. Pengujian Black Box Halaman Penyakit

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman penyakit. Pengujian dan hasil uji *black box* halaman penyakit pada Tabel 4.5.

Tabel 4. 5 Pengujian Black Box Menu Halaman Penyakit.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Create Penyakit	Muncuk tampilan untuk menambah data penyakit	Sukses
Klik Search	Search untuk mencari nama penyakit	Sukses
Klik tombol Edit	Muncuk tampilan untuk ngubah data	Sukses
Klik tombol Delete	Fungsi untuk menghapus data	Sukses
Klik tombol Next	Fungsi untuk melihat halaman data penyakit yang selanjutnya	Sukses

Klik tombol Previous	Fungsi untuk kembali ke halaman data penyakit yang sebelumnya	Sukses
-------------------------	---	--------

#### f. Pembuatan Program Halaman Gejala

Halaman gejala adalah halaman yang ada pada admin setelah admin melakukan halaman login lalu masuk ke halaman dashboard, setelah itu klik menu gejala. Halaman penyakit ini menampilkan semua data gejala yang ada. pada halaman gejala ini menampilkan menu-menu yang ada pada sistem yaitu menu *create* penyakit, *search*, *edit*, dan *delete*. Berikut potongan *source code* halaman gejala :

```
<div class="content">
    <div class="grid-system">
        <div class="row-center">
            <div class="card">
                @if (session('success'))
                    <div class="alert alert-success alert-dismissible">
                        <a class="close" data-dismiss="alert" href="#">X</a>
                        <h4 class="alert-heading">Sukses!</h4>
                        {{ session('success') }}
                    </div>
                @endif
                @if (session('edit'))
                    <div class="alert alert-success alert-dismissible">
                        <a class="close" data-dismiss="alert" href="#">X</a>
                        <h4 class="alert-heading">Sukses!</h4>
                        {{ session('edit') }}
                    </div>
                @endif
                <div class="container">
                    <div class="header">
                        <h3>Gejala</h3>
                    </div>
                    <div class="col-md-12">

                        <div class="btn-group" role="group" style="padding-right: 50px; box-shadow: none !important; float: right;">
                            <a class="btn btn-primary waves-effect" href="{{ route('gejala.create') }}"> Create gejala</a>
                        </div><br><br>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        <div class="body table-responsive">
            <table class="table table-bordered
table-striped" id="gejala">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>nama</th>
                        <th>Action</th>
                    </tr>
                </thead>
                </table>
            </div>
        </div>
    </div>
</div>
<script type="text/javascript">
// $("#reload").click(function () {
//     table.ajax.reload();
// });
$(function () {
    $.ajaxSetup({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-
token"]').attr('content')
        }
    });
    window.table = $('#gejala').DataTable({
        processing: true,
        serverSide: true,
        ajax: "{{ route('gejala.index') }}",
        columns: [
            {data: 'kode_gejala', name: 'kode_gejala'},
            {data: 'nama_gejala', name: 'nama_gejala'},
            {data: 'action', name: 'action'},
        ],
        orderable: false, searchable: false,
    });
});

function deleteData(kode_penyakit){
    var csrf_token = $('meta[name="csrf-
token"]').attr('content');
    swal({
        title: 'Are you sure?',
        text: "You won't be able to revert this!",
        type: 'warning',
        showCancelButton: true,
        cancelButtonColor: '#d33',
        confirmButtonColor: '#3085d6',
        confirmButtonText: 'Yes, delete it!'
    }).then(function(isConfirm) {
        if (isConfirm) {
            $.ajax({
                url : "{{ route('gejala.destroy', '') }}" +
'/' + kode_penyakit,
                type : "DELETE",
            })
        }
    })
}

```

```

        data : { '_method' : '_DELETE', '_token' :
    csrf_token},
        success : function(data) {
            table.ajax.reload();
            swal({
                title: 'Success!',
                text: data.message,
                type: 'success',
                timer: '1500'
            })
        },
        error : function () {
            swal({
                title: 'Oops...',
                text: data.message,
                type: 'error',
                timer: '1500'
            })
        }
    });
}
}).catch(swal.noop);
}

</script>
@endsection

```

## 1. Pengujian White Box Halaman Gejala

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman gejala berupa tampilan halaman data gejala yang terdiri dari menu *create* penyakit, *search*, *edit*, *delete*, dan *next*. Halaman gejala telihat pada Gambar 4.13.

Gejala		
		<a href="#">Create gejala</a>
ID	nama	Action
1	Perdarahan hebat yang terus menerus keluar dari vagina setelah persalinan	<a href="#">Edit</a> <a href="#">Delete</a>
2	Darah yang keluar berwarna merah terang	<a href="#">Edit</a> <a href="#">Delete</a>
3	keluarnya bekuan darah yang berukuran lebih besar	<a href="#">Edit</a> <a href="#">Delete</a>
4	Cepat Lelah	<a href="#">Edit</a> <a href="#">Delete</a>
5	Cemas	<a href="#">Edit</a> <a href="#">Delete</a>
6	Konsentrasi menurun	<a href="#">Edit</a> <a href="#">Delete</a>
7	Jantung berdebar	<a href="#">Edit</a> <a href="#">Delete</a>
8	Sesak napas	<a href="#">Edit</a> <a href="#">Delete</a>
9	Turunnya tekanan darah	<a href="#">Edit</a> <a href="#">Delete</a>
10	Sulit dan nyeri saat buang air kecil	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 10 of 32 entries

Previous 1 2 3 4 Next

Gambar 4. 13 Tampilan Halaman Gejala..

## 2. Pengujian Black Box Halaman Gejala

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman gejala. Pengujian dan hasil uji *black box* halaman gejala pada Tabel 4.6.

Tabel 4. 6 Pengujian Black Box Menu Halaman Gejala.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Create Gejala	Muncuk tampilan untuk menambah data gejala	Sukses
Klik Search	Search untuk mencari nama gejala	Sukses
Klik tombol Edit	Muncuk tampilan untuk ngubah data	Sukses
Klik tombol Delete	Fungsi untuk menghapus data	Sukses
Klik tombol Next	Fungsi untuk melihat halaman data gejala yang selanjutnya	Sukses
Klik tombol Previous	Fungsi untuk kembali ke halaman data gejala yang sebelumnya	Sukses

## g. Pembuatan Program Halaman Pengetahuan

Halaman pengetahuan adalah halaman yang ada pada admin setelah admin melakukan halaman login lalu masuk ke halaman dashboard, setelah itu klik menu pengetahuan. Halaman pengetahuan ini menampilkan data pengetahuan yang ada. pada halaman pengetahuan ini menampilkan menu-menu yang ada pada sistem yaitu menu *create* pengetahuan, *search*, *edit*, *delete*, *next*, dan *previous*. Berikut potongan *source code* halaman penyakit:

```
<div class="content">
    <div class="grid-system">
        <div class="row-center">
            <div class="card">
```

```

        @if (session('success'))
            <div class="alert alert-success alert-dismissible">
                <a class="close" data-dismiss="alert"
                    href="#">X</a>
                <h4 class="alert-heading">Sukses!</h4>
                {{ session('success') }}
            </div>
        @endif
        @if (session('edit'))
            <div class="alert alert-success alert-dismissible">
                <a class="close" data-dismiss="alert"
                    href="#">X</a>
                <h4 class="alert-heading">Sukses!</h4>
                {{ session('edit') }}
            </div>
        @endif
        <div class="container">
            <div class="header">
                <h3>Pengetahuan</h3>
            </div>
            <div class="col-md-12">

                <div class="btn-group" role="group" style="padding-right: 50px; box-shadow: none !important; float: right;">
                    <a class="btn btn-primary waves-effect" href="{{ route('pengetahuan.create') }}">Create Pengetahuan</a>
                </div>

                </div><br><br>
                <div class="body table-responsive">
                    <table class="table table-bordered table-striped" id="pengetahuan">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>penyakit</th>
                                <th>gejala</th>
                                <th>mb</th>
                                <th>md</th>
                                <th>Action</th>
                            </tr>
                        </thead>
                        </table>
                </div>
            </div>
        </div>
    </div>
</div>
<script type="text/javascript">
```

```

//    $("#reload").click(function () {
//        table.ajax.reload();
//    });
$(function () {
    $.ajaxSetup({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-
token"]').attr('content')
        }
    });
    window.table = $('#pengetahuan').DataTable({
        processing: true,
        serverSide: true,
        ajax: "{{ route('pengetahuan.index') }}",
        columns: [
            {data: 'kode_pengetahuan', name: 'kode_pengetahuan'},
            {data: 'penyakit.nama_penyakit', name: 'penyakit.nama_penyakit'},
            {data: 'gejala.nama_gejala', name: 'gejala.nama_gejala'},
            {data: 'mb', name: 'mb'},
            {data: 'md', name: 'md'},
            {data: 'action', name: 'action'},
        ],
        orderable: false, searchable: false,
    });
});

function deleteData(kode_pengetahuan) {
    var csrf_token = $('meta[name="csrf-
token"]').attr('content');
    swal({
        title: 'Are you sure?',
        text: "You won't be able to revert this!",
        type: 'warning',
        showCancelButton: true,
        cancelButtonColor: '#d33',
        confirmButtonColor: '#3085d6',
        confirmButtonText: 'Yes, delete it!'
    }).then(function(isConfirm) {
        if (isConfirm) {
            $.ajax({
                url : "{{ route('pengetahuan.destroy', '') }}" + '/' + kode_pengetahuan,
                type : "DELETE",
                data : {'_method' : '_DELETE', '_token' : csrf_token},
                success : function(data) {
                    table.ajax.reload();
                    swal({
                        title: 'Success!',
                        text: data.message,
                    });
                }
            });
        }
    });
}

```

```

        type: 'success',
        timer: '1500'
    })
},
error : function () {
    swal({
        title: 'Oops...',
        text: data.message,
        type: 'error',
        timer: '1500'
    })
})
);
}) .catch(swal.noop);
}
</script>
@endsection

```

## 1. Pengujian White Box Halaman Pengetahuan

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman pengetahuan berupa tampilan halaman data pengetahuan yang terdiri dari menu *create* pengetahuan, *search*, *edit*, *delete*, dan *next*. Halaman pengetahuan telihat pada Gambar 4.14.

Pengetahuan						
						<a href="#">Create Pengetahuan</a>
Show <input type="button" value="10"/> entries <input type="text" value="Search:"/>						
ID	penyakit	gejala	mb	md	Action	
1	Preeklamsia	Sesak napas	1	0.6	<a href="#">Edit</a>	<a href="#">Delete</a>
2	Embolik Paru	Kulit ujung-ujung jari dan bibir membiru	0.8	0.6	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Embolik Paru	Pingsan	0.6	0.4	<a href="#">Edit</a>	<a href="#">Delete</a>
4	Embolik Paru	Merasa pusing	0.6	0.4	<a href="#">Edit</a>	<a href="#">Delete</a>
5	Embolik Paru	Gelisah	0.4	0.4	<a href="#">Edit</a>	<a href="#">Delete</a>
6	Embolik Paru	Demam	0.6	0.4	<a href="#">Edit</a>	<a href="#">Delete</a>

Showing 1 to 6 of 6 entries

Previous 1 Next

Activate Win  
Cara Cetakan

Gambar 4. 14 Tampilan Halaman Pengetahuan.

## 2. Pengujian Black Box Halaman Pengetahuan

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman pengetahuan. Pengujian dan hasil uji *black box* halaman pengetahuan pada Tabel 4.7.

Tabel 4. 7 Pengujian Black Box Menu Halaman Pengetahuan.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Create Pengetahuan	Muncuk tampilan untuk menambah data pengetahuan	Sukses
Klik Search	Search untuk mencari nama pengetahuan	Sukses
Klik tombol Edit	Muncuk tampilan untuk ngubah data	Sukses
Klik tombol Delete	Fungsi untuk menghapus data	Sukses
Klik tombol Next	Fungsi untuk melihat halaman data pengetahuan yang selanjutnya	Sukses
Klik tombol Previous	Fungsi untuk kembali ke halaman data pengetahuan yang sebelumnya	Sukses

## h. Pembuatan Program Halaman Data User

Halaman penyakit adalah halaman yang ada pada admin setelah admin melakukan halaman login lalu masuk ke halaman dashboard, setelah itu klik menu *user*. Halaman *user* ini menampilkan semua data *user* yang ada. pada halaman *user* ini menampilkan menu-menu yang ada pada sistem yaitu menu *create user, search, edit, delete, next, dan previous*. Berikut potongan *source code* halaman *user* :

```
<div class="content">
    <div class="grid-system">
```

```

<div class="row-center">
    <div class="card">
        @if (session('success'))
            <div      class="alert      alert-success      alert-
dismissible">
                <a      class="close"      data-dismiss="alert"
href="#">X</a>
                <h4 class="alert-heading">Sukses!</h4>
                {{ session('success') }}
            </div>
        @endif
        @if (session('edit'))
            <div      class="alert      alert-success      alert-
dismissible">
                <a      class="close"      data-dismiss="alert"
href="#">X</a>
                <h4 class="alert-heading">Sukses!</h4>
                {{ session('edit') }}
            </div>
        @endif
        <div class="container">
            <div class="header">
                <h3>User</h3>
            </div>
            <div class="col-md-12">

                <div      class="btn-group"      role="group"
style="
                    padding-right: 50px;
                    box-shadow: none !important;
                    float: right;
                ">
                    <a      class="btn      btn-primary      waves-
effect" href="{{ route('users.create') }}"> Create New User</a>
                </div>

                </div><br><br>
                <div class="body table-responsive">
                    <table class="table table-bordered table-
striped" id="users-table">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Name</th>
                                <th>Email</th>
                                <th>Action</th>
                            </tr>
                        </thead>
                        <tbody>
                            <tr>
                                <td>1</td>
                                <td>John Doe</td>
                                <td>john.doe@example.com</td>
                                <td>
                                    <a href="#">Edit</a>
                                    <a href="#">Delete</a>
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
<script type="text/javascript">
    $("#reload").click(function () {
        table.ajax.reload();
    });
    $(function () {
        $.ajaxSetup({
            headers: {

```

```

        'X-CSRF-TOKEN':           $('meta[name="csrf-
token"]').attr('content')
    }
});
window.table = $('#users-table').DataTable({
    processing: true,
    serverSide: true,

    ajax: "{{ route('users.index') }}",
    columns: [
        {data: 'DT_RowIndex', name: 'DT_RowIndex'},
        {data: 'name', name: 'name'},
        {data: 'email', name: 'email'},
        {data: 'action', name: 'action', orderable:
false, searchable: false},
    ]
});
});

function deleteData(id) {
    var csrf_token = $('meta[name="csrf-
token"]').attr('content');
    swal({
        title: 'Are you sure?',
        text: "You won't be able to revert this!",
        type: 'warning',
        showCancelButton: true,
        cancelButtonColor: '#d33',
        confirmButtonColor: '#3085d6',
        confirmButtonText: 'Yes, delete it!'
    }).then(function(isConfirm) {
        if (isConfirm) {
            $.ajax({
                url : "{{ route('users.destroy', '') }}/" +
id,
                type : "DELETE",
                data : {'_method' : '_DELETE', '_token' :
csrf_token},
                success : function(data) {
                    table.ajax.reload();
                    swal({
                        title: 'Success!',
                        text: data.message,
                        type: 'success',
                        timer: '1500'
                    })
                },
                error : function () {
                    swal({
                        title: 'Oops...',
                        text: data.message,
                        type: 'error',
                        timer: '1500'
                    })
                }
            });
        }
    }).catch(swal.noop);
}
</script>

```

### 1. Pengujian White Box Halaman Data User

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman Data User berupa tampilan halaman data *user* yang terdiri dari menu *create* pengetahuan, *search*, *edit*, *delete*, dan *next*. Halaman pengetahuan telihat pada Gambar 4.15.

User					
				Create New User	
Show 10 entries		Search:			
ID	Name	Email	Action		
1	admin	admin@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>		
2	Rina	rina123@yahoo.com	<a href="#">Edit</a> <a href="#">Delete</a>		
3	nia safitri	safitri@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>		
4	nia	nia@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>		

Showing 1 to 4 of 4 entries

Previous 1 Next

Gambar 4. 15 Tampilan Halaman Data User.

### 2. Pengujian Black Box Halaman Data User

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman data *user*. Pengujian dan hasil uji *black box* halaman data *user* pada Tabel 4.8.

Tabel 4. 8 Pengujian Black Box Menu Halaman Data User.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Create New User	Muncuk tampilan untuk menambah data User dan admin	Sukses
Klik Search	Search untuk mencari nama user dan admin	Sukses
Klik tombol Edit	Muncuk tampilan untuk ngubah data	Sukses
Klik tombol Delete	Fungsi untuk menghapus data	Sukses

Klik tombol Next	Fungsi untuk melihat halaman data pengetahuan yang selanjutnya	Sukses
Klik tombol Previous	Fungsi untuk kembali ke halaman data pengetahuan yang sebelumnya	Sukses

### i. Pembuatan Program Halaman Diagnosa Penyakit

Halaman diagnosa penyakit adalah halaman setelah *user* melakukan login, setelah itu masuk ke halaman diagnosa penyakit. Pada halaman diagnosa penyakit ini menampilkan menu pilihan dan hasil. Berikut potongan *source code* halaman diagnosa penyakit :

```
<div class="clearfix"></div>
    <div style="font-size:16px;">
        <div>
            <div class='alert alert-success alert-dismissible'>
                <button type='button' class='close' data-dismiss='alert' aria-hidden='true'>x</button>
                <h4><i class='icon fa fa-exclamation-triangle'></i>Perhatian !</h4>
                Silahkan memilih gejala sesuai dengan kondisi anda, anda dapat memilih kepastian kondisi anda dari pasti tidak sampai pasti ya, jika sudah tekan tombol proses
                <i class='fa fa-search-plus'></i> di bawah untuk melihat hasil.
            </div>
        </div>
    </div> <!-- /end content -->
</div>

</div>
<div>&nbsp;</div>
<div class="timeline-img">
    <div class="container">
        <form action="{{ route('hasil.cf') }}" method="post">
            {{ csrf_field() }}
            <input type="hidden" name='email' value="{{$email}}'
            <table class='table table-bordered table-striped'>
                <tbody>
                    <tr>
                        <th width=8%>No</th>
                        <th width=10%>Kode</th>
                        <th>Gejala yang dialami (keluhan)</th>
                        <th width=20%>Pilihan</th>

```

```

        </tr>
        @php $i=1 @endphp
        @foreach($gejala as $g)
        <tr>
            <td>@php echo $i @endphp</td>
            <td>G {{ sprintf('%02d',
$g['kode_gejala']) }}</td>
            <td>
                <select name="kondisi[]" id="sl' . $i . '" class="opsikondisi"><option data-id="0" value="0">Pilih jika sesuai</option>
                @foreach($kondisi as $k)
                    <option data-id="{{ $k->id }}" value="{{ $g['kode_gejala'] . '_' . $k->id }}> {{ $k->kondisi }}</option>
                @endforeach
                </select>
            </td>
        </tr>
        @php $i++ @endphp
    @endforeach
    </tbody>
</table>
<button class='fa fa-search pull-right col-md-2' type=submit title='Klik disini untuk melihat hasil diagnosa'>
    hasil
</button>
</form>
</div> <!-- /end content -->
</div>

<div class="container">
<br>
    <table class='table table-bordered table-striped'>
        <th width=8%>No</th>
        <th width=10%>Kode</th>
        <th>Gejala yang dialami (keluhan)</th>
        <th width=20%>Pilihan</th>
        @php $i=1 @endphp
        @php $a=0 @endphp
        @foreach($argejala ?? [] as $key => $value)
        <tr>
            <td>@php echo $i @endphp</td>
            <td>G {{ sprintf('%02d', $key) }}</td>
            <td>{{ $gejala[$a]['nama_gejala'] }}</td>
            <td>{{ $get[$value]['kondisi'] }}</td>
        </tr>
    
```

## 1. Pengujian White Box Halaman Diagnosa Penyakit

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman diagnosa penyakit yang terdiri dari menu pilihan dan hasil. Halaman diagnose penyakit terlihat pada Gambar 4.16 dan 4.17.



No	Kode	Gejala yang dialami (keluhan)	Pilihan
1	G 01	Pendarahan hebat yang terus menerus keluar dari vagina setelah persalinan	Pilih jika sesuai
2	G 02	Darah yang keluar berwarna merah terang	Pilih jika sesuai
3	G 03	keluarnya bekuan darah yang berukuran lebih besar	Pilih jika sesuai
4	G 04	Cepat lelah	Pilih jika sesuai
5	G 05	Cemas	Pilih jika sesuai
6	G 06	Konsentrasi menurun	Pilih jika sesuai
7	G 07	Jantung berdebar	Pilih jika sesuai
8	G 08	Sesak nafas	Pilih jika sesuai
9	G 09	Turunnya tekanan darah	Pilih jika sesuai
10	G 10	Sulit dan nyeri saat buang air kecil	Pilih jika sesuai
11	G 11	Urin keruh atau berdarah	Pilih jika sesuai
12	G 12	Nyeri perut bagian bawah	Pilih jika sesuai

Gambar 4. 16 Tampilan Halaman Diagnosa Penyakit.

14	G 14	nyeri wkw	Pilih jika sesuai
15	G 15	Kehilangan atau sakit kepala	Pilih jika sesuai
16	G 16	Jarang buang air kecil	Pilih jika sesuai
17	G 17	Berat badan naik mendadak	Pilih jika sesuai
18	G 18	Mau pinggan	Pilih jika sesuai
19	G 19	Pembengkakan pada kakik	Pilih jika sesuai
20	G 20	Pandangan kabur	Pilih jika sesuai
21	G 21	Menggigil	Pilih jika sesuai
22	G 22	Sakit perut bagian atas	Pilih jika sesuai
23	G 23	Demam	Pilih jika sesuai
24	G 24	Gelisah	Pilih jika sesuai
25	G 25	Pingsan	Pilih jika sesuai
26	G 26	Kulit ujung-ujung jari dan bibir memburu	Pilih jika sesuai
27	G 27	Merasa pusing	Pilih jika sesuai
28	G 28	Perasaan sedih / menangis tanpa alasan	Pilih jika sesuai
29	G 29	Sulit tidur (Insomnia)	Pilih jika sesuai
30	G 30	Lemah	Pilih jika sesuai
31	G 31	Cepat marah	Pilih jika sesuai
32	G 32	Gampang tersinggung	Pilih jika sesuai

hasil

Gambar 4. 17 Tampilan Halaman Diagnosa Penyakit.

## 2. Pengujian Black Box Halaman Diagnosa Penyakit

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman diagnosa penyakit. Pengujian dan hasil uji *black box* halaman diagnosa penyakit pada Tabel 4.9.

Tabel 4. 9 Pengujian Black Box Menu Halaman Diagnosa Penyakit.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik menu Pilihan	Muncuk tampilan untuk memilih kondisi	Sukses
Klik tombol Hasil	Fungsi untuk menampilkan hasil diagnose penyakit pasca melahirkan	Sukses

#### j. Pembuatan Program Halaman Untuk Memasukkan Data Penyakit

Halaman untuk memasukkan data penyakit adalah halaman ketika masuk ke halaman penyakit setelah itu mengklik menu “Create Penyakit” maka muncul halaman untuk memasukkan data penyakit. Pada halaman untuk memasukkan data penyakit ini menampilkan menu-menu yang ada pada sistem yaitu menu choose file, back, dan menu submit. Berikut potongan *source code* halaman memasukan data penyakit :

```
<form id="form_validation" method="POST" action="{{ route('penyakit.store') }}"
@else
<form id="form_validation" method="POST" action="{{ route('penyakit.update', $penyakit->kode_penyakit) }}"
@endif
{{ csrf_field() }}
```

```

<div class="form-group form-float">
<div class="form-line">
<label class="form-label">Nama Penyakit</label>
<input type="text" class="form-control" name="nama_penyakit" value="{{ $penyakit->nama_penyakit }}" minlength="3">
</div>
<div class="help-info"></div>
</div>
<div class="form-group form-float">
<div class="form-line">
<label class="form-label">Detail</label>
<textarea id="mymce" name="det_penyakit" class="form-control" style="height: 100px;">
</div>
</div>
```

```

value="{{old('det_penyakit', $penyakit->det_penyakit)}}"
placeholder="Detail">>{{$penyakit->det_penyakit}}</textarea>

```

```

                </div>
                <div class="help-info"></div>
            </div>
            <div class="form-group form-float">
                <div class="form-line">
                    <label class="form-label">Saran</label>
                    <textarea id="mce" name="srn_penyakit" value="{{old('srn_penyakit', $penyakit->srn_penyakit)}}"
                        placeholder="Saran">>{{$penyakit->srn_penyakit}}</textarea>
                </div>
                </div>
                <div class="help-info"></div>
            </div>
            <br>
            <div class="form-group form-float">
                <div class="form-line">
                    <label class="form-label">Image</label>
                    <div class="fileDiv btn btn-info btn-flat">
                        <i class="fa fa-upload"></i>
                        <input type='file' id="image" name="image" onchange="loadFile(image)"/>
                    </div>
                    
                    <div class="help-info"><span style="color:red;">*</span>only Jpg, png format </div>
                </div>
            </div>

            <div class="col-md-12"></div>
            <div class="col-md-12"></div>
            <div class="col-md-12">

                <a class="btn btn-warning waves-effect" href="{{route('penyakit.index')}}" style="float: left;" type="submit">
                    BACK
                </a>
                <button class="btn btn-primary waves-effect" type="submit" style="float: right;">
                    SUBMIT
                </button>
            </div>
        </form>
    
```

## 1. Pengujian White Box Halaman Untuk Memasukkan Data Penyakit

Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman untuk memasukkan data penyakit yang terdiri dari tiga menu. Halaman untuk memasukkan data penyakit pada Gambar 4.18.

Gambar 4. 18 Tampilan Halaman Memasukkan Data Penyakit.

## 2. Pengujian *Black Box* Halaman Untuk Memasukkan Data Penyakit

Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman untuk memasukkan data penyakit. Pengujian dan hasil uji *black box* halaman untuk memasukkan data penyakit pada Tabel 4.10.

Tabel 4. 10 Pengujian Halaman Untuk Memasukkan Data Penyakit.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Choose File	Fungsi untuk memasukkan Gambar Penyakit	Sukses
Klik tombol Back	Kembali pada halaman penyakit	Sukses
Klik tombol Submit	Fungsi untuk menyimpan data penyakit jika sudah mengisi nama penyakit, detail, saran, dan gambar.	Sukses

	Jika nama penyakit, detail, saran, dan gambar belum diisi	Gagal
--	---	-------

#### k. Pembuatan Program Halaman Memasukan Data Gejala

Halaman untuk memasukkan data gejala adalah halaman ketika masuk ke halaman gejala setelah itu mengklik menu “Create Gejala” maka muncul halaman untuk memasukkan data gejala. Pada halaman untuk memasukkan data gejala ini menampilkan menu-menu yang ada pada sistem yaitu menu back, dan menu submit. Berikut potongan *source code* halaman memasukan data gejala :

```

<form id="form_validation" method="POST" action="{{ route('gejala.update', $gejala->kode_gejala) }}"
enctype="multipart/form-data">
    <input type="hidden" name="_method" value="PUT">
    @endif
    {{ csrf_field() }}

    <div class="form-group form-float">
        <div class="form-line">
            <label class="form-label">Nama gejala</label>
            <input type="text" class="form-control" name="nama_gejala" value="{{$gejala->nama_gejala}}" minlength="3">
        </div>
        <div class="help-info"></div>
    </div>

    <div class="col-md-12"></div>
    <div class="col-md-12"></div>
    <div class="col-md-12">

        <a class="btn btn-warning waves-effect" href="{{route('gejala.index')}}" style="float: left;" type="submit">
            BACK
        </a>
        <button class="btn btn-primary waves-effect" type="submit" style="float: right;">
            SUBMIT
        </button>
    </div>
</form>
```

- Pengujian White Box Halaman Untuk Memasukkan Data Gejala  
Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman untuk memasukkan data gejala yang terdiri dari 2 menu. Halaman untuk memasukkan data gejala pada Gambar 4.19.

Nama gejala

BACK SUBMIT

Gambar 4. 19 Tampilan Halaman Untuk Memsukkan Data Gejala.

- Pengujian Black Box Halaman Untuk Memasukkan Data Gejala  
Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman untuk memasukkan data gejala. Pengujian dan hasil uji *black box* halaman untuk memasukkan data gejala pada Tabel 4.11.

Tabel 4. 11 Pengujian Halaman Untuk Memasukkan Data Gejala.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Back	Kembali pada halaman gejala	Sukses
Klik tombol Submit	Fungsi untuk menyimpan data penyakit jika sudah mengisi nama gejala	Sukses
	Jika nama gejala belum diisi	Gagal

- Pembuatan Program Halaman Untuk Menambahkan Data Pengetahuan  
Halaman untuk memasukkan data pengetahuan adalah halaman ketika masuk ke halaman pengetahuan setelah itu mengklik menu “Create Pengetahuan” maka muncul halaman untuk memasukkan data pengetahuan. Pada halaman

untuk memasukkan data pengetahuan ini menampilkan menu-menu yang ada pada sistem yaitu menu back, dan menu submit. Berikut potongan *source code* halaman memasukan data penyakit :

```

<form      id="form_validation"      method="POST"      action="{{
route('pengetahuan.update', $pengetahuan->kode_pengetahuan)  }}"
enctype="multipart/form-data">
    <input type="hidden" name="_method" value="PUT">
    @endif
    {{ csrf_field() }}

    <div class='alert alert-success alert-dismissible'>
        <button      type='button'      class='close'      data-
dismiss='alert' aria-hidden='true'></button>
        <h4><i      class='icon      fa      fa-
exclamation-triangle'></i>Petunjuk Pengisian Pakar !</h4>
        Silahkan pilih gejala yang sesuai
        dengan penyakit yang ada, dan berikan <b>nilai kepastian (MB &
        MB)</b> dengan cakupan sebagai berikut:<br><br>
        <b>1.0</b>                               (Pasti
        Ya) &nbsp;&nbsp; | &nbsp;&nbsp;<b>0.8</b>           (Hampir
        Pasti) &nbsp;&nbsp; | <br>
        <b>0.6</b>                               (Kemungkinan
        Besar) &nbsp;&nbsp; | &nbsp;&nbsp;&nbsp;<b>0.4</b>
        (Mungkin) &nbsp;&nbsp; | <br>
        <b>0.2</b>                               (Hampir
        Mungkin) &nbsp;&nbsp; | &nbsp;&nbsp;&nbsp;<b>0.0</b>   (Tidak Tahu atau
        Tidak Yakin) &nbsp;&nbsp; | <br><br>
        <b>CF(Pakar) = MB - MD</b><br>
        MB : Ukuran kenaikan kepercayaan (measure of increased belief)
        MD : Ukuran kenaikan ketidakpercayaan (measure of increased
        disbelief) <br> <br>
        <b>Contoh:</b><br>
        Jika kepercayaan <b>(MB)</b> anda terhadap gejala
        keluarnya bekuan darah yang berukuran lebih besar
        adalah <b>0.8 (Hampir Pasti)</b><br>
        Dan ketidakpercayaan <b>(MD)</b> anda terhadap
        keluarnya bekuan darah yang berukuran lebih besar
        adalah <b>0.2 (Hampir Mungkin)</b><br><br>
        <b>Maka:</b> CF(Pakar) = MB - MD (0.8 - 0.2) =
        <b>0.6</b> <br>
        Dimana nilai kepastian anda terhadap keluarnya
        bekuan darah yang berukuran lebih besar adalah
        <b>0.6 (Kemungkinan Besar)</b>
    </div>
    <br><br>
    <div class="input-group">
        <span class="input-group-
addon">
            <i      class="material-
icons">
                Penyakit
            </i>
        </span>
        <select      class="form-
control show-tick"      name="kode_penyakit" >
            @foreach($penyakit as
            $value)

```

```

@if($value-
>nama_penyakit == $pengetahuan->penyakit['nama_penyakit'])
    <option value="{!!
$value->kode_penyakit      !!}" selected="selected">{!!
strtoupper($value->nama_penyakit) !!}</option>
    @else
        <option value="{!!
$value->kode_penyakit      !!}">{!!    strtoupper($value->nama_penyakit)
}!</option>
    @endif
@endforeach

</select>
</div><br>

<div class="input-group">
    <span class="input-group-
addon">
        <i    class="material-
icons">
            Gejala
        </i>
    </span>
    <select      class="form-
control show-tick" name="kode_gejala" >
        @foreach($gejala   as
$value)
            @if($value-
>nama_gejala == $pengetahuan->gejala['nama_gejala'])
                <option value="{!!
$value->kode_gejala !!}" selected="selected">{!!    strtoupper($value-
>nama_gejala) !!}</option>
            @else
                <option value="{!!
$value->kode_gejala      !!}">{!!    strtoupper($value->nama_gejala)
}!</option>
            @endif
        @endforeach

        </select>
    </div><br>

    <div class="form-group form-float
col-md-12">
        <div class="col-md-6">
            <div class="form-line">
                <label    class="form-
label">MB</label>
                <input      type="text"
class="form-control"    name="mb"    value="{!!
{$pengetahuan->mb} !!}"
minlength="3">
            </div>
            <div      class="help-
info"></div>
        </div>
        <div class="col-md-6">
            <div class="form-line">
                <label    class="form-
label">MD</label>
            </div>
        </div>
    </div>

```

```

<input type="text"
class="form-control" name="md" value="{{ $pengetahuan->md }}"
minlength="3">
</div>
<div class="help-
info"></div>
</div>

<div class="col-md-12"></div>
<div class="col-md-12"></div>
<div class="col-md-12">

<a class="btn btn-warning waves-effect" href="{{ route('gejala.index') }}" style="float: left;" type="submit">
    BACK
</a>
<button class="btn btn-primary waves-effect" type="submit" style="float: right;">
    SUBMIT
</button>
</div>
</form>

```

1. Pengujian White Box Halaman Untuk Memasukkan Data Pengetahuan  
Pengujian *white box* dilakukan dengan melakukan pengujian kode program (*coding*). Hasil *compile* halaman untuk memasukkan data pengetahuan yang terdiri dari 2 menu. Halaman untuk memasukkan data pengetahuan pada Gambar 4.20.

Gambar 4. 20 Tampilan Halaman Untuk Memasukkan Data pengetahuan.

2. Pengujian Black Box Halaman Untuk Memasukkan Data Pengetahuan  
Pengujian *black box* dilakukan dengan menguji setiap aktivitas yang terdapat pada halaman untuk memasukkan data pengetahuan. Pengujian

dan hasil uji *black box* halaman untuk memasukkan data pengetahuan pada Tabel 4.12.

Tabel 4. 12 Pengujian Halaman Untuk Memasukkan Data Pengetahuan.

Aktivitas	Realisasi Yang di Harapkan	Hasil
Klik tombol Back	Kembali pada halaman pengetahuan	Sukses
Klik tombol Submit	Fungsi untuk menyimpan data pengetahuan jika sudah mengisi nama penyakit, gejala, nilai MB, dan nilai MD.	Sukses
	Jika nama penyakit, gejala, nilai MB, dan nilai MD belum diisi	Gagal

## 4.2 Pembahasan

Pemanfaatan metode *Certainty Factor* untuk diagnosa penyakit pasca melahirkan berbasis web ini telah melalui proses pengujian perangkat lunak dengan metode *black box*. Hasilnya dari pengujian dari fitur-fitur dapat berjalan dengan normal dan bekerja sesuai dengan rancangan ditandai saat sistem menerima inputan yang salah ditandatai dengan pesan kesalahan ketika sistem menerima inputan yang tidak sesuai dengan aturan input yang sudah diterapkan pada setiap textbox. Dengan hasil tersebut maka dapat dilakukan bahwa tujuan pembuatan sistem pemanfaatan metode *Certainty Factor* untuk diagnosa penyakit pasca melahirkan berbasis web sudah tercapai.

Dilihat dari awal memilih kondisi gejala yang ada pada aplikasi pemanfaat metode *Certainty Factor* diagnosa penyakit pasca melahirkan kemudian sistem melakukan proses diagnosa penyakit, kemudian muncul hasil jenis penyakit yang diderita oleh pasien, persentasi penyakit yang diderita, detail penyakit, dan muncul saran dari penyakit yang diderita. Setelah itu tampil daftar Riwayat, daftar Riwayat tersebut berisikan data pasien yang telah

melakukan diagnosa penyakit pasca melahirkan, yang isinya kode pasien, nama pasien, tanggal, nama penyakit, dan nilai cf, kemudian daftar cf tersebut bisa dicetak untuk diberikan kepada pasien atau disimpan.

Sistem ini diharapkan dapat diterapkan sehingga dapat membantu bagi petugas atau perawat agar lebih cepat dalam mendiagnosa penyakit pasca melahirkan dan hasil diagnosa dapat langsung diberikan kepada dokter atau bidan dan dapat segera ditangani.

## **BAB V**

### **KESIMPULAN DAN PENELITIAN SELANJUTNYA**

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian yang sudah dilakukan mengenai pemanfaatan metode *Certainty Factor* untuk mendiagnosa penyakit pasca melahirkan memiliki keunggulan yang berada dari penelitian sebelumnya dimana sistem yang dibangun bersifat dinamis sehingga data basis pengetahuan dapat ditambahkan dan diperbaharui oleh admin. Hasil dari algoritma CF adalah 0,20% dengan memasukan variabel yaitu cepat lelah, tekanan darah naik sampai 140%/90mmHg, jarang buang air kecil, berat badan naik mendadak, pembekakkan pada kaki, pandangan kabur, dan sakit perut bagian bawah, maka hasil tersebut adalah pembagian dari beberapa penyakit.

#### **5.2 Penelitian Selanjutnya**

Dari penelitian yang telah dilakukan, mengingat berbagai keterbatasan yang dialami penulis mengenai masalah pemikiran dan waktu. disarankan untuk penelitian selanjutnya agar dapat mencoba menelaah dan mengkaji dengan lebih dalam mengenai metode yang lain. Memperbanyak data penyakit dan gejala agar mendapatkan hasil yang maksimal. Dalam memelihara keakuratan data pada aplikasi ini maka perlu dilakukan proses *uprate* basis pengetahuan secara berkala. Dan sistem yang dibangun ini masih memiliki banyak kekurangan, baik dari segi fungsional maupun data yang dimiliki.

## **JADWAL PENYUSUNAN TUGAS AKHIR**

## DAFTAR PUSTAKA

- [1] N. Setiawan and . P. . N. Andono , "Sistem Pakar Diagnosa Penyakit Seorang Ibu Pasca Melahirkan," *DRAFT JURNAL*, p. 2, 2016.
- [2] Edah, "Faktor Predisposisi Terjadinya Perdarahan Postpartum Di Rumah Sakit Umum Daerah Muntilan Tahun 2018," Universitas Muhammadiyah Magelang, Magelang, 2019.
- [3] d. . A. Haryono, "Infeksi Postpartum," SehatQ, 29 October 2019. [Online]. Available: <https://www.sehatq.com/penyakit/infeksi-postpartum>. [Accessed 13 Maret 2020].
- [4] D. Kurniasari and F. Arifandini, "Hubungan Usia, Paritas Dan Diabetes Mellitus Pada Kehamilan Dengan Kejadian Preeklamsia Pada Ibu Hamil Di Wilayah Kerja Puskesmas Rumbia Kabupaten Lampung Tengah Tahun 2016," *Jurnal Kesehatan Holistik*, vol. 9, p. 143, 2015.
- [5] R. Halodoc, "Emboli pada Ibu Setelah Melahirkan Bisa Berakibat Fatal," Halodoc, 25 Oktober 2018. [Online]. Available: <https://www.halodoc.com/emboli-pada-ibu-setelah-melahirkan-bisa-berakibat-fatal>. [Accessed 13 Maret 2020].
- [6] . . R. Fadli, "Ibu Baru Bisa Alami Baby Blues Syndrome, Ini Cara Mengatasinya," Halodoc, 24 Januari 2020. [Online]. Available: <https://www.halodoc.com/kenali-atasi-baby-blues-syndrome-pada-ibu>. [Accessed 13 Maret 2020].
- [7] P. Hasan , E. W. Sholeha , Y. N. Tetik and Kusrini, "Sistem Pakar Diagnosa Penyakit Kolesterol dan Asam Urat Menggunakan Metode Certainty Factor," *Jurnal Ilmiah SISFOTEKNIKA*, vol. 9, no. 1, p. 2, 2019.

- [8] Y. K. Sari, D. Kartini and M. , "Implementasi Algoritma SAW (Simple Additive Weighting) Dempster Shafer Pada Diagnosa Awal Postpartum Depression," *CESS (Journal of Computer Engineering System and Science)*, vol. 3, no. 1, p. 1, 2018.
- [9] A. Hasbiyanor and B. , "Sistem Pakar Diagnosa Keluhan Selama Masa Kehamilan Menggunakan Metode Certainty Factor Berbasis Web," *JUTISI*, vol. 6, no. 1, p. 2, 2017.
- [10] T. S. M.Kom, E. M. M.Kom and D. V. Suhartono, *Kecerdasan Buatan*, Yogyakarta: C.V Andi OFFSET , 2011.
- [11] R. R. Fanny, N. A. Hasibuan and . E. Buulolo, "Perancangan Sistem Pakar Diagnosa Penyakit ASIDOSIS Tubulus Renalis Menggunakan Metode Certainty Factor Dengan Penulusuran Forward Chaining," *MEDIA INFORMATIKA BUDIDARMA*, vol. 1, no. 1, p. 14, 2017.
- [12] D. P. 2, "Dosen Pendidikan," 5 Oktober 2019. [Online].
- [13] G. Merry, "Majalah Pendidikan," 3 November 2019. [Online]. Available: <https://majalahpendidikan.com/definisi-penyakit-dan-jenis-penyakit-yang-mengancam-manusia/>.
- [14] Marmi, Asuhan Neonatus, Bayi, Balita dan Anak Prasekolah, Yogyakarta: Pustaka Pelajar, 2012.
- [15] "Lusa Hari Pasti Lebih Cerah," 23 Agustus 2009. [Online]. Available: <https://lusa.afkar.id/asuhan-sayang-ibu-sebagai-kebutuhan-dasar-persalinan>.
- [16] A. Veratamala, "Apa Yang Terjadi Setelah Ibu Melahirkan".
- [17] D. Rahmawati, "Waspadai Kematian Akibat Perdarahan Post Partum," SehatQ, 04 May 2019. [Online]. Available:

- <https://www.sehatq.com/artikel/tanda-pendarahan-post-partum-setelah-melahirkan>. [Accessed 13 Maret 2020].
- [18] L. Tan, "Sepsis," SehatQ, 09 December 2018. [Online]. Available: <https://www.sehatq.com/penyakit/sepsis>. [Accessed 13 Maret 2020].
- [19] R. Halodoc, "Preeklamsia," halodoc, 22 November 2019. [Online]. Available: <https://www.halodoc.com/kesehatan/preeklamsia>. [Accessed 13 Maret 2020].
- [20] C. I. Restyana and F. Adiesti, "Kejasian Baby Blues Pada Ibu Primipara Di RSUD Bangil Pasuruan," *Hospital Majapahit*, vol. 6, p. 33, 2014.
- [21] S. . A. Putri and E. P. Saputra, "Perancangan Aplikasi Sistem Pakar Diagnosa Awal Kanker Reproduksi Wanita Dengan Metode Certainty Factor," *MEDIA INFORMATIKA BUDIDARMA*, vol. 2, p. 64, 2018.
- [22] Y. Yudhanto and H. . A. Prasetyo, Mudah Menguasai Framework Laravel, Jakarta: PT Exex Media Komputindo, 2019.
- [23] J. Enterprise, PHP untuk Pemograman Pemula, Jakarta: PT Elex Media Komputindo, 2019.
- [24] A. "Dunia Ilmu Komputer," 22 Juli 2019. [Online]. Available: <https://www.duniailmu.com/pengertian-dan-fungsi-php-dalam-pemograman-web/>.
- [25] D. Mediana and A. I. Nurhidayat, "Rancangan Bangun Aplikasi Helpdesk (A-DESK) Berbasis Web Minggunakan Framework Laravel (Studi Kasus Di PDAM Sueya Sembada Kota Surabaya)," *Jurnal Manajemen Informatika*, vol. 8, p. 76, 2018.
- [26] B. Hermanto, . M. Yusman and N. , "Sistem Informasi Manajemen Keuangan Pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel," *Jurnal Komputasi*, vol. 7, p. 19, 2019.

- [27] A. HRV, "Laragon". 5 Mei 2019.
- [28] T. Fujimaru, "Laragon, Alternatif XAMPP Terbaik di Windows". 26 April 2018.
- [29] A. Fahrudin, B. E. Purnama and B. K. Riasti, "Pembangunan Sistem Informasi Layanan Haji Berbasis Web Pada Kelompok Bimbingan Ibadah Haji Ar Rohmsn Mbrur Kudus," *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, vol. 3, p. 36, 2011.
- [30] B. S. Ir. and H. I. P. Ir., M.Eng, Pemograman Web Dengan HTML, Bandung : Informatika Bandung, 2012.
- [31] B. S. Ir. and P. H. I. Ir.M.Eng, Pemograman Web dengan HTML, Bandung: Informatika Bandung, 2012.
- [32] R. Y. W, "Ekstensi dan Tema Visual Studio Code yang Saya Gunakan". 3 Februari 2019.
- [33] P. P. Abdullah, "Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Metode Certainty Factor Berbasis Android," Lampung, 2016.
- [34] N. "<https://informatikalogi.com/>," 10 April 2017. [Online]. Available: <https://informatikalogi.com/pengertian-flowchart-dan-jenis-jenisnya/>. [Accessed 2 Maret 2020].
- [35] R. A. S and M. Shalahuddin, Rekayasa Perangkat Lunak, Bandung: Informatika Bandung, 2016.
- [36] A. Nugroho, Rekayasa Perangkat Lunak Menggunakan UML dan Java, Yogyakarta: C.V ANDI OFFSET , 2009.
- [37] C. Tristianto, "Penggunaan Metode Waterfall Untuk Pengembangan Sistem Monotoring Dan Evaluasi Pembangunan Pedesaan," *Jurnal Teknologi Informasi ESIT*, vol. XII, p. 13, 2018.

- [38] W. H. Khatimi and R. Maulana, "Sistem Pakar Diagnosa Jenis Gangguan Jiwa Skizofrenia Menggunakan Kombinasi Metode Forward Chaining Dan Certainty Factor," *JTIULM (Jurnal Teknologi Informasi Universitas Lambung Mangkurat)*, vol. 02, p. 2, 2017.

## RIWAYAT HIDUP PENULIS



**Nia Sapitri** adalah nama penulis Laporan Tugas Akhir ini. Penulis lahir dari orang tua bapak bernama Masrani dan ibu bernama Fitriah sebagai anak pertama dari 2 (dua) bersaudara. Penulis ini di lahirkan di desa Sungai Malang, Kecamatan Amuntai Tengah, Kabupaten Hulu Sungai Utara, Kalimantan Selatan pada rabu 28 Oktober 1998. Penulis menempuh Pendidikan dimulai dari TK Aisyah (lulus tahun 2004), melanjutkan ke SDN Sungai Malang 5 (lulus tahun 2010), melanjutkan ke SMP Negeri 4 Amuntai (lulus tahun 2013), melanjutkan ke SMA Negeri 1 Amuntai (lulus tahun 2016) hingga akhirnya untuk menempuh masa kuliah ke Program Studi S1 Informatika Fakultas Teknik Universitas Muhammadiyah Banjarmasin (lulus tahun 2020). Penulis juga aktif di dunia organisasi mulai dari Himpunan Mahasiswa Teknik Informatika Universitas Muhammadiyah Banjarmasin (HMTI UMBJM) menjabat sebagai bendahara umum HMTI, Paduan Suara, dan Unit Kegiatan Mahasiswa (UKM) Sosial.

Dengan ketekunan, motivasi tinggi untuk belajar dan berusaha penulis telah berhasil menyelesaikan Laporan Tugas Akhir ini. Semoga dengan penulisan Laporan TA ini mampu memberikan kontribusi positif bagi Program Studi S1 Informatika, Fakultas Teknik, maupun Universitas Muhammadiyah Banjarmasin.

Akhir kata penulis mengucapkan rasa syukur yang sebesar-besarnya atas terselesaikannya Laporan TA yang berjudul “**Pemanfaatan Metode Certainty Factor Untuk Diagnosa Penyakit Pasca Melahirkan**”.

Penulis