

**SISTEM PAKAR DIAGNOSIS PENYAKIT PADA  
BALITA BERBASIS *WEB***

**SKRIPSI**



**Oleh:  
Richardus Damar  
140210171**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
TAHUN 2019**

**SISTEM PAKAR DIAGNOSIS PENYAKIT PADA  
BALITA BERBASIS *WEB***

**SKRIPSI**

**Untuk memenuhi salah satu syarat  
guna memperoleh gelar sarjana**



**Oleh  
Richardus Damar  
140210171**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
TAHUN 2019**

## **PERNYATAAN**

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian penulis sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini Penulis buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka penulis bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 11 February 2019  
Yang membuat pernyataan,

Richardus Damar  
140210171

**SISTEM PAKAR *DIAGNOSIS* PENYAKIT PADA  
BALITA BERBASIS *WEB***

**Oleh  
Richardus Damar  
140210171**

**SKRIPSI  
Untuk memenuhi salah satu syarat  
guna memperoleh gelar Sarjana**

**Telah disetujui oleh dosen pembimbing pada tanggal  
seperti tertera di bawah ini**

**Batam, 11 February 2019**

**Hotma Pangaribuan, S.Kom., M.SI.  
Pembimbing**

## ***ABSTRAK***

Sebagian balita lebih rentan terhadap penyakit karena kondisi geografis Indonesia yang berada di daerah tropis menjadikan variasi mikroorganisme penyebab penyakit yang lebih beragam. Oleh sebab itu gejala penyakit dan cara penanganannya perlu dikenali seperti batuk, suara serak, suhu tinggi, gangguan pada saluran pencernaan. Penyakit yang sering di derita oleh balita antara lain *Dengue haemorrhagic fever*, kejang demam, infeksi saluran pernapasan, demam *typhoid* dan diare. Maka di dunia kesehatan dibutuhkan sebuah sistem untuk membantu dokter dalam mendiagnosa sebuah penyakit agar penanganannya cepat di atasi. Oleh karena itu dibutuhkan sistem pakar yang berfungsi untuk membantu mendiagnosa, memberikan solusi dan cara penanggulangan pada penyakit balita. Pada penelitian ini dirancang sebuah sistem pakar menggunakan metode inferensi yang bekerja mundur kearah kondisi awal. *Output* dari sistem pakar ini adalah sebuah program untuk mendiagnosa penyakit pada balita berdasarkan gejala-gejala penyakit dan solusi yang di input oleh *user* kedalam program. Penelitian ini akan dilakukan pengujian pada aplikasi sistem pakar diagnosis penyakit pada balita, yang nantinya akan menghasilkan kesimpulan dan solusi. Menggunakan aplikasi berbasis *web* dengan bahasa pemrograman PHP dengan penyimpanan MySQL. Sehingga menghasilkan sebuah aplikasi yang dapat digunakan untuk *diagnosis* penyakit pada balita.

**Kata Kunci:** Penyakit Balita, Sistem Pakar, *Backward Chaining*, *Web*.

## **ABSTRAK**

*Some toddlers are more susceptible to disease because of the geographical conditions in Indonesia in the tropics, making the variety of micro-organisms that cause disease more diverse. Therefore the symptoms of the disease and how to handle them need to be recognized such as coughing, hoarseness, high temperature, disturbances in the digestive tract. Diseases that are often suffered by toddlers include Denque haemorrhagic fever, febrile seizures, respiratory infections, typhoid fever and diarrhea. So in the world of health a system is needed to help doctors diagnose a disease so that treatment can be quickly resolved. Therefore we need an expert system that serves to help diagnose, provide solutions and ways of dealing with underfive diseases. In this study an expert system was designed using the inference method that works backwards towards the initial condition. The output of this expert system is a program to diagnose diseases in toddlers based on the symptoms of the disease and solutions that are input by the user into the program. This research will be tested on an expert system application for diagnosing diseases in infants, which will later produce conclusions and solutions. Using a web-based application with the PHP programming language with MySQL storage. So as to produce an application that can be used to diagnose diseases in infants.*

**Keywords: Toddler Disease, Expert System, Backward Chaining, Web.**

## KATA PENGANTAR

Segala puji bagi Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Bapak Andi Maslan, S.T., M.SI. selaku Ketua Program Studi Teknik Informatika Universitas Putera Batam.
3. Bapak Hotma Pangaribuan, S.Kom., M.SI selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Kedua orang tua penulis tercinta Darius Dat dan Donata Nanus yang selalu memberikan doanya serta dukungannya kepada penulis hingga skripsi ini selesai.

6. Adek dan Kakak saya Heribertus Aban, Yester Namur, Marselina Amul yang selalu mendukung menyemangati saya dalam pembuatan skripsi.
7. Terima kasih juga kepada Zakaria, Jahya, Jawandi, Bobby, Daus, Jundilla, Purnama, Edi telah menyemangati saya dalam menyelesaikan skripsi ini.
8. Seluruh teman-teman yang telah membantu dalam penyelesaian skripsi.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu memberikan hal yang baik untuk kita semua.

Batam, 11 February 2019

Penulis

## DAFTAR ISI

	Halaman
HALAMAN PERNYATAAN .....	iii
HALAMAN PENGESAHAN.....	iv
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah.....	4
1.3 Pembatasan Masalah .....	4
1.4 Perumusan Masalah .....	5
1.5 Tujuan Penelitian .....	5
1.6 Manfaat Penelitian .....	6
1.6.1 Manfaat Bagi Objek .....	6
1.6.2 Manfaat Peneliti .....	6
BAB II KAJIAN PUSTAKA .....	
2.1 Teori Dasar.....	7
2.1.1 Kecerdasan Buatan atau <i>Artificial Intelligence</i> (AI).....	7
2.1.2 <i>FuzzyLogic</i> (Logika Fuzzy).....	8
2.1.3 Jaringan Saraf Tiruan .....	11
2.1.4 Sistem Pakar ( <i>Expert System</i> ) .....	12
2.1.4.1 Manfaat Sistem Pakar .....	13
2.1.4.2 Kekurangan Sistem Pakar .....	14
2.1.4.3 Ciri-ciri Sistem Pakar .....	14
2.1.5 Struktur Sistem Pakar.....	15
2.1.6 Teknik Inferensi .....	17
2.1.6.1 <i>Backward Chaining</i> .....	18
2.1.7 Tabel dan Pohon Keputusan.....	18
2.2 Variabel Penelitian .....	20
2.3 <i>Software</i> Pendukung .....	28
2.3.1 PHP .....	29
2.3.1.1 Definisi PHP .....	29
2.3.1.2 Kelebihan PHP .....	29
2.3.2 XAMPP .....	30
2.3.2.1 Definisi XAMPP .....	30
2.3.2.2 <i>APACHE</i> .....	31
2.3.2.3 <i>PHP MyAdmin</i> .....	32

2.3.3 MySQL.....	33
2.3.3.1 Definisi MySQL.....	33
2.3.3.2 Cara Kerja <i>MySQL</i> .....	33
2.3.4 Star UML .....	35
2.3.4.1 Definisi <i>Star UML</i> .....	35
2.3.5 <i>Notepad ++</i> .....	35
2.3.6 <i>UML</i> .....	36
2.4 Penelitian Terdahulu .....	44
2.5 Kerangka Pemikiran.....	47
<b>BAB III METODOLOGI PENELITIAN</b>	
3.1 Desain Penelitian.....	49
3.2 Teknik Pengumpulan Data.....	51
3.3 Operasional Variabel.....	52
3.4 Perancangan Sistem .....	53
3.4.1 Desain Basis Pengetahuan .....	53
3.4.2 Desain <i>UML(Unified Modeling Language)</i> .....	64
3.4.3 Desain Antarmuka.....	90
3.5 Lokasi dan Jadwal Penelitian.....	100
3.5.1 Lokasi.....	100
3.5.2 Jadwal Penelitian.....	101
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
4.1 Hasil Penelitian .....	102
4.1.1 Implementasi Sistem .....	102
4.2 Pembahasan.....	113
4.2.1 Pengujian validasi sistem .....	113
<b>BAB V SIMPULAN DAN SARAN</b>	
5.1 Simpulan .....	116
5.2 Saran.....	117

DAFTAR PUSTAKA

DAFTAR RIWAYAT HIDUP

LAMPIRAN I FOTO WAWANCARA

LAMPIRAN II FORM WAWANCARA

LAMPIRAN III KODING PROGRAM

## DAFTAR TABEL

	Halaman
<b>Tabel 2. 1</b> Tabel Keputusan.....	19
<b>Tabel 2. 2</b> <i>Use Case Diagram</i> .....	38
<b>Tabel 2. 3</b> <i>Activity Diagram</i> .....	39
<b>Tabel 2. 4</b> <i>Squence Diagram</i> .....	40
<b>Tabel 2. 5</b> <i>Diagram Class</i> .....	43
<b>Tabel 3. 1</b> Variabel dan Indikator.....	52
<b>Tabel 3. 2</b> Data Penyakit pada Balita .....	53
<b>Tabel 3. 3</b> Data Gejala Penyakit pada Balita .....	54
<b>Tabel 3. 4</b> Tabel Data Aturan .....	56
<b>Tabel 3. 5</b> Tabel keputusan gejala dan penyakit pada balita .....	57
<b>Tabel 3. 6</b> Tabel Solusi.....	60
<b>Tabel 3. 7</b> Jadwal Penelitian.....	102
<b>Tabel 4. 1</b> Tabel Pengujian Validasi Sistem pada Menu <i>User</i> .....	113
<b>Tabel 4. 2</b> Tabel Pengujian Validasi Sistem pada Menu <i>Admin</i> .....	114

## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Struktur Sistem Pakar .....	15
<b>Gambar 2.2</b> Pohon Keputusan .....	19
<b>Gambar 2.3</b> <i>Denque haermorhagic fever</i> .....	21
<b>Gambar 2.4</b> Penyakit Kejang Demam .....	23
<b>Gambar 2.5</b> Penyakit ISPA.....	24
<b>Gambar 2.6</b> Penyakit Thypoid.....	26
<b>Gambar 2.7</b> Penyakit Diare .....	27
<b>Gambar 2.8</b> Logo PHP .....	29
<b>Gambar 2.9</b> Logo XAMPP .....	30
<b>Gambar 2.10</b> Logo PhpMyAdmin.....	32
<b>Gambar 2.11</b> MySql .....	33
<b>Gambar 2.12</b> Logo <i>Star UML</i> .....	35
<b>Gambar 2.13</b> Kerangka Berpikir .....	48
<b>Gambar 3.1</b> Desain Penelitian .....	49
<b>Gambar 3.2</b> Pohon Keputusan .....	64
<b>Gambar 3.3</b> <i>Use Case Diagram</i> .....	65
<b>Gambar 3.4</b> <i>Activity Diagram login admin dan user</i> .....	66
<b>Gambar 3.5</b> <i>Activity Diagram Data Penyakit</i> .....	67
<b>Gambar 3.6</b> <i>Activity Diagram Data Gejala</i> .....	68
<b>Gambar 3.7</b> <i>Activity Diagram Solusi</i> .....	70
<b>Gambar 3.8</b> <i>Activity Diagram Menu Relasi</i> .....	71
<b>Gambar 3.9</b> <i>Acitivity Diagram Menu Logout</i> .....	72
<b>Gambar 3.10</b> <i>Activity Diagram Tampilan User</i> .....	73
<b>Gambar 3.11</b> <i>Activity Diagram Daftar</i> .....	74
<b>Gambar 3.12</b> <i>Activity Diagram Kelola User</i> .....	75
<b>Gambar 3.13</b> <i>Activity Diagram Hasil Diagnosa</i> .....	76
<b>Gambar 3.14</b> <i>Sequence Diagram Login Admin dan User</i> .....	77
<b>Gambar 3.15</b> <i>Sequence Diagram Logout Admin dan User</i> .....	78
<b>Gambar 3.16</b> <i>Sequence Diagram Penyakit</i> .....	79
<b>Gambar 3.17</b> <i>Sequence Diagram Kelola Gejala</i> .....	80
<b>Gambar 3.18</b> <i>Sequence Diagram Solusi</i> .....	81
<b>Gambar 3.19</b> <i>Sequence Diagram Relasi</i> .....	82
<b>Gambar 3.20</b> <i>Sequence Diagram Kelola User</i> .....	83
<b>Gambar 3.21</b> <i>Sequence Diagram Diagnosa</i> .....	85
<b>Gambar 3.22</b> <i>Sequence Diagram Diagnosa</i> .....	86
<b>Gambar 3.23</b> <i>Class Diagram</i> .....	87
<b>Gambar 3.24</b> <i>Desain Database</i> .....	89
<b>Gambar 3.25</b> <i>Form Login</i> .....	90
<b>Gambar 3.26</b> <i>Form Penyakit</i> .....	90
<b>Gambar 3.27</b> <i>Form Tambah Penyakit</i> .....	91

<b>Gambar 3.28</b> <i>Form Edit Penyakit</i> .....	92
<b>Gambar 3.29</b> <i>Form Gejala</i> .....	92
<b>Gambar 3.30</b> <i>Form Tambah Gejala</i> .....	93
<b>Gambar 3.31</b> <i>Form Edit Gejala</i> .....	93
<b>Gambar 3.32</b> <i>Form Solusi</i> .....	94
<b>Gambar 3.33</b> <i>Form Tambah Solusi</i> .....	94
<b>Gambar 3.34</b> <i>Form Edit Solusi</i> .....	95
<b>Gambar 3.35</b> <i>Form Relasi</i> .....	95
<b>Gambar 3.36</b> <i>Form Relasi Gejala</i> .....	96
<b>Gambar 3.37</b> <i>Form Relasi Solusi</i> .....	96
<b>Gambar 3.38</b> <i>Form Kelola User</i> .....	97
<b>Gambar 3.39</b> <i>Form Tambah User</i> .....	98
<b>Gambar 3.40</b> <i>Form Diagnosa</i> .....	98
<b>Gambar 3.41</b> <i>Form Profil</i> .....	99
<b>Gambar 3.42</b> <i>Form Data Diagnosa</i> .....	99
<b>Gambar 3.43</b> <i>Form Halaman Home</i> .....	100
<b>Gambar 4.1</b> <i>Menu Login User</i> .....	103
<b>Gambar 4.2</b> <i>Menu Home User</i> .....	103
<b>Gambar 4.3</b> <i>Tampilan Menu Diagnosa User</i> .....	104
<b>Gambar 4.4</b> <i>Tampilan Menu Diagnosa User</i> .....	104
<b>Gambar 4.5</b> <i>Tampilan Menu Data Diagnosa</i> .....	105
<b>Gambar 4.6</b> <i>Tampilan Menu Data Diagnosa</i> .....	105
<b>Gambar 4.7</b> <i>Tampilan Menu Bantuan User</i> .....	106
<b>Gambar 4.8</b> <i>Tampilan Menu Profil User</i> .....	106
<b>Gambar 4.9</b> <i>Tampilan Menu Login Admin</i> .....	107
<b>Gambar 4.10</b> <i>Tampilan Menu Home Admin</i> .....	108
<b>Gambar 4.11</b> <i>Menu Penyakit</i> .....	108
<b>Gambar 4.12</b> <i>Menu Gejala</i> .....	109
<b>Gambar 4.13</b> <i>Menu Solusi</i> .....	110
<b>Gambar 4.14</b> <i>Menu Relasi</i> .....	110
<b>Gambar 4.15</b> <i>Menu Kelola Admin</i> .....	111
<b>Gambar 4.16</b> <i>Menu Laporan</i> .....	112
<b>Gambar 4.17</b> <i>Menu Profil Admin</i> .....	112

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penggunaan teknologi komputer saat ini sudah semakin berkembang sesuai dengan kebutuhan manusia yang semakin beragam. Komputer pada awalnya digunakan hanya sebagai alat hitung. Seiring dengan perkembangan zaman, komputer dikembangkan lagi untuk membantu pekerjaan manusia di berbagai bidang, misalnya saja dalam bidang kesehatan, ekonomi, pendidikan, pertanian dan sebagainya. Sebagai contoh, dengan adanya komputer segala kegiatan dapat dilakukan dengan cepat dan mampu mengurangi resiko kesalahan. Di dalam perkembangan ilmu komputer, para ahli dibidangnya mencoba untuk menciptakan suatu sistem yang diharapkan dapat memiliki kemampuan memecahkan suatu permasalahan layaknya seorang ahli. Hal inilah yang mendorong lahirnya konsep sistem pakar.

Menurut Ibu Hartuti S.ST selaku bidan Rumah Sakit Awal Bros Batam balita lebih sering mengalami penyakit *denque haermorrhagic fever*, kejang demam, infeksi saluran pernapasan, demam *thypoid* dan diare, karena kondisi indonesia yang berada di daerah tropis menyebabkan penyakit yang lebih beragam. Maka gejala penyakit dan cara penangannya perlu diketahui seperti batuk, suara serak, suhu tinggi, gangguan pada saluran pencernaan. Untuk dapat mencegah timbulnya komplikasi lebih jauh orang tua harus banyak memeriksa

gejala-gejala penyakit yang timbul pada balita dan sering melakukan pengobatan dengan dokter untuk mencegah kesalahan dalam pemberian pertolongan pertama kepada balita. Maka di dunia kesehatan dibutuhkan sebuah sistem untuk membantu dokter/bidan dalam mendiagnosa sebuah penyakit agar penanganannya cepat di atasi, sistem pakar berbasis *web* banyak digunakan kalangan masyarakat umum.

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia. Sistem pakar telah banyak dikembangkan di berbagai bidang seperti dibidang kesehatan, khususnya dibidang mendiagnosa penyakit pada balita. Karena itu penyakit pada balita terjadi karena gangguan kesehatan terhadap anak maka mereka lebih mempercayakannya kepada pakar atau dokter ahli yang sudah mengetahui lebih banyak tentang kesehatan, tanpa memperdulikan apakah gangguan tersebut masih dalam tingkat rendah atau kronis. Namun dengan kemudahan adanya para pakar atau dokter ahli, terkadang terdapat pula kelemahannya seperti jam kerja (praktek) terbatas dan banyaknya pasien hingga harus menunggu antrian yang panjang dengan biaya yang tidak sedikit, penderita penyakit pada balita tentu harus banyak memperhatikan kesehatan tubuhnya, tetapi terkadang juga para orang tua kesulitan dalam menangani masalah penyakit tersebut dan cara pengobatannya karena keterbatasan informasi. Didalam penelitian ini dibangun sebuah sistem pakar untuk membantu para penderita

penyakit pada balita dalam menangani dan mengetahui jenis penyakit pada balita yang di alami dengan mengimplementasikan pengetahuan seorang pakar kedalam sebuah *web* dan akan menghasilkan manfaat seperti memberikan informasi yang tepat dan cepat di akses, sehingga membantu meringankan pekerjaan pakar dalam menjalankan tugasnya.

Agar tujuan penelitian penyakit pada balita dapat dibuktikan, maka dibutuhkan sebuah metode *backward chaining*, sangat baik digunakan untuk sistem diagnosa penyakit pada balita karena prosesnya dilakukan dari kesimpulan yang menjadi solusi dari permasalahan yang dihadapi dengan melakukan rangkaian test dengan cara mengisi kusioner dan wawancara kepada seorang pakar.

Berdasarkan kesimpulan jurnal penelitian (Dewi, Mulyani, & Restianie, 2016) diperoleh fakta jurnal bahwa aplikasi sistem pakar ini dapat mendiagnosa suatu penyakit anak (balita) yang sering diderita berdasarkan gejala yang dialami, kemudian dapat memberikan informasi penyakit tersebut beserta solusinya. Dengan diberikan *interface* yang *user friendly* maka aplikasi sistem pakar ini dapat mudah dipelajari dan mudah digunakan untuk kalangan orang awam, sehingga dapat mempermudah dalam mendiagnosa penyakit. Berdasarkan kesimpulan jurnal penelitian (Trianto, Informatika, Pamulang, & Selatan-indonesia, 2018) Dengan menggunakan aplikasi sistem diagnosa penyakit diare ini, para pengguna khususnya orang tua dapat dengan mudah memperoleh informasi penanganan pertama yang harus dilakukan terkait dengan hasil diagnosa.

Penelitian ini mengimplementasikan suatu analisa secara sistematis berbasis *web* menggunakan metode *backward chaining*. Berdasarkan permasalahan yang telah diuraikan diatas, diangkatlah sebuah penelitian dengan judul “**Sistem Pakar Diagnosis Penyakit Pada Balita Berbasis Web**” dengan penelitian ini diharapkan dapat membantu masyarakat mencari informasi tentang penyakit balita.

## 1.2 Identifikasi Masalah

Adapun identifikasi masalah penelitian adalah sebagai berikut :

1. Kurangnya pemahaman dan pengetahuan pasien dalam mengidentifikasi penyakit pada balita.
2. Dibutuhkan biaya, waktu untuk mendatangkan dokter/bidan dalam menangani penyakit pada balita.
3. Belum adanya sistem yang membantu pengguna dalam mendiagnosa penyakit pada balita berbasis *web*.

## 1.3 Pembatasan Masalah

Batasan masalah penelitian sistem pakar diagnosa penyakit pada balita ini meliputi:

1. Sistem pakar hanya membahas penyakit pada balita yaitu *denque haermorrhagic fever*, kejang demam, infeksi saluran pernapasan, demam *thypoid* dan diare.

2. Pakar pada penelitian ini adalah Ibu Hartuti S.ST selaku bidan Rumah Sakit Awal Bros Batam.
3. Metode yang digunakan adalah *backward chaining* sebagai inferensi.
4. Sistem pakar berbasis *web* menggunakan *software* antara lain: *Star UML*, *XAMPP*, *PhpMyAdmin*, *PHP*, *MySQL*, *Notepad++*.

#### **1.4 Perumusan Masalah**

Berdasarkan latar belakang diatas penulis dapat menyimpulkan perumusan masalah sebagai berikut :

1. Bagaimana implementasi sistem pakar dalam mendiagnosa penyakit pada balita?
2. Bagaimana implementasi sistem pakar dalam mendiagnosa penyakit pada balita dengan metode *backward chaining*?
3. Bagaimana sistem pakar memberikan solusi dan cara penanggulangan yang tepat dalam mendiagnosa penyakit balita?

#### **1.5 Tujuan Penelitian**

Adapun tujuan dari mendiagnosa penyakit pada balita adalah sebagai berikut:

1. Untuk membangun sistem pakar dalam mendiagnosa penyakit pada balita.
2. Untuk membangun metode *backward chaining* dalam mendiagnosa penyakit pada balita.

3. Untuk memberikan solusi dan cara penanggulangnya.

## **1.6 Manfaat Penelitian**

Peneliti mengharapkan dapat memberikan manfaat bagi objek maupun manfaat bagi peneliti kepada pembaca, manfaat tersebut antara lain:

### **1.6.1 Manfaat Teoritis**

Adapun manfaat bagi objek dari penelitian ini adalah:

1. Penelitian ini diharapkan dapat membantu orang tua dalam mengakses sebuah informasi pengetahuan yang bermanfaat untuk pengembangan penelitian sistem pakar penyakit pada balita dengan menggunakan sistem berbasis *web*.
2. Mengetahui jenis penyakit pada balita, gejala dan solusi dalam mendiagnosa penyakit pada balita.

### **1.6.2 Manfaat Praktis**

Adapun manfaat bagi peneliti dari penelitian ini adalah:

1. Dapat membuat program aplikasi sesuai dengan permasalahan yang dihadapi.
2. Penulis dapat mengetahui cara membuat suatu sistem pakar berbasis *web* dan mengimplementasikan kepada masyarakat umum yang belum mengetahuinya.

## **BAB II KAJIAN PUSTAKA**

### **2.1. Teori Dasar**

Menurut (Dr.Sudaryono, 2015, p. 15) Deskripsi teori dalam suatu penelitian merupakan uraian sistematis tentang teori (dan bukan sekedar pendapat pakar atau penulis buku) dan hasil penelitian yang relevan dengan *variable* yang diteliti. Berapa jumlah kelompok teori yang dikemukakan atau dideskripsikan akan tergantung pada luasnya permasalahan dan secara teknis tergantung pada jumlah *variable* yang diteliti. Bila dalam suatu penelitian terdapat tiga *variable* independen dan satu dependen, maka kelompok teori yang perlu dideskripsikan ada empat kelompok teori, yaitu kelompok teori yang berkenaan dengan *variable* independen dan satu dependen , oleh karena itu semakin banyak *variable* yang diteliti, maka akan semakin banyak teori yang dikemukakan.

#### **2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)***

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 1) kecerdasan buatan berasal dari bahasa inggris “*Artificial intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud merujuk pada mesin yang mampu

berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Berdasarkan definisi ini, maka kecerdasan buatan menawarkan media maupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemrograman dan eksekusinya dapat dibuktikan pada komputer nyata. Program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik, kecerdasan buatan memungkinkan komputer untuk berpikir atau menalar dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan, pengalaman, dan proses pembelajaran serta dapat digunakan sebagai acuan dimasa-masa yang akan datang (Sutojo, Mulyanto, & Suhartono, 2011, p. 3).

### **2.1.2 *Fuzzy Logic (Logika Fuzzy)***

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 211) logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk di implementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisi data, dan *system control*. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak ataupun kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”.

Bila dibandingkan dengan logika konvensional, kelebihan logika *fuzzy* adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan persamaan matematik yang rumit. Beberapa alasan yang dapat diutarakan mengapa kita menggunakan logika *fuzzy* di antaranya adalah mudah dimengerti, memiliki toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi *non linear* yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan, dapat bekerja sama dengan teknik-teknik kendali secara konvensional, dan didasarkan pada bahasa alami (Sutojo, Mulyanto, & Suhartono, 2011, p. 212).

Untuk memahami logika *fuzzy*, sebelumnya perhatikan konsep himpunan *fuzzy*. Himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. Linguistik, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya dingin, sejuk, panas.
2. Numeris, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40 dan sebagainya.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Sutojo, Mulyanto, & Suhartono, 2011, p. 233)

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)

- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (Minimum)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

## 2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan *output* diperlukan 4 tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (Minimum) dan komposisi antar-*rule* menggunakan fungsi *MAX*(Maximum) dengan menghasilkan himpunan *fuzzy* baru
- d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)

## 3. Metode Sugeno

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Dalam metode ini, *output* sistem berupa konstanta atau persamaan linier. Dalam inferensinya, metode Sugeno menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (Minimum)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

### 2.1.3 Jaringan Syaraf Tiruan (JST)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 284) Jaringan saraf tiruan mempunyai kemampuan yang luar biasa untuk mendapatkan informasi dari data yang rumit atau tidak tepat, mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan, dapat belajar dari pengalaman, mampu mengakuisisi pengetahuan walaupun tidak ada kepastian, mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*), mampu memilih suatu input data ke dalam kategori tertentu yang sudah ditetapkan (klasifikasi), mampu menggambarkan suatu proyek secara keseluruhan walaupun hanya diberikan sebagian data dari objek tersebut (asosiasi), mempunyai kemampuan mengolah data-data input tanpa harus mempunyai target (*self organizing*), serta mampu menemukan jawaban terbaik sehingga mampu meminimalisasi fungsi biaya (optimasi).

Kelebihan-kelebihan yang diberikan oleh jaringan saraf tiruan antara lain :

- a. Belajar *adaftive*: kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
- b. *Self-Organization*: sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.

- c. *Real Time Operation*: perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

#### **2.1.4 Sistem Pakar (*Expert System*)**

Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul campuran yang tak dikenal, *XCON & XSEL* untuk membantu konfigurasi sistem komputer besar, *SOPHIE* untuk analisis sirkuit elektronik, *prospector* digunakan di bidang geologi untuk membantu mencari dan menemukan *deposite*, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya. Istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge* (Sutojo, Mulyanto, & Suhartono, 2011, p. 159).

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.

#### 2.1.4.1 Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya (Sutojo, Mulyanto, & Suhartono, 2011, p. 161).

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas *system computer*. Integrasi *system* pakar dengan *system computer* lain membuat *system* lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan *system computer* konvensional, *system* pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan *system* pakar tetap akan memberikan jawabannya.

10. Bisa digunakan sebagai media pelengkap dan pelatihan. Pengguna pemula yang bekerja dengan system pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

#### **2.1.4.2 Kekurangan Sistem Pakar**

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100 % bernilai benar.

#### **2.1.4.3 Ciri-Ciri Sistem Pakar**

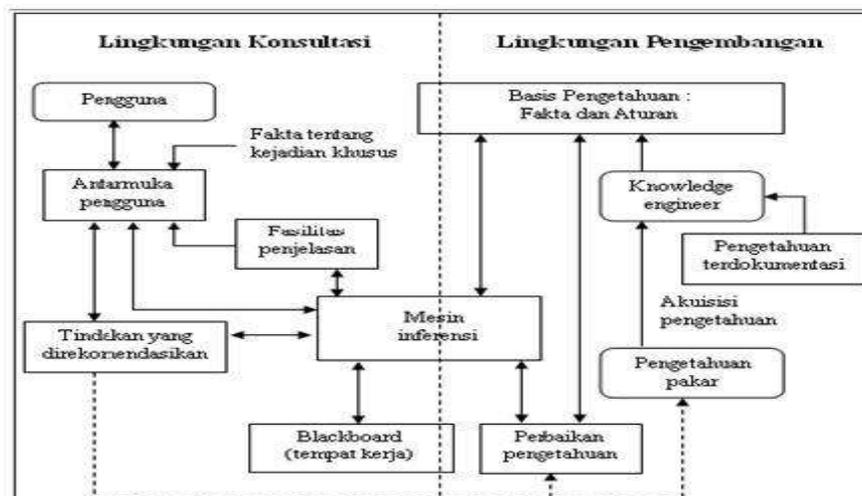
Ciri ciri dari sistem pakar (Sutojo, Mulyanto, & Suhartono, 2011:162) adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/rule tertentu.
5. Mudah di modifikasi.

6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna

### 2.1.5 Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapat pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. (Sutojo, Mulyanto, & Suhartono, 2011, p. 166).



**Gambar 2.1** Struktur Sistem Pakar

Sumber : (Sutojo, Mulyanto, & Suhartono, 2011, p. 167)

Berikut penjelasan dari subsistem yang ada pada struktur sistem pakar pada Gambar 2.1:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu.

2. Basis pengetahuan

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule*.

3. Mesin Inferensi

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi dan kesimpulan.

4. Daerah kerja (*Blackboard*)

*Blackboard* merupakan area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard* yaitu rencana, agenda dan solusi.

5. Antarmuka pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dengan sistem pakar. Sistem perbaikan pengetahuan Kemampuan memperbaiki pengetahuan dari

seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

#### 6. Pengguna (*user*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada. Pengetahuan yang dimasukkan ke dalam sistem pakar disajikan dalam bentuk yang dapat dimengerti dan diterima oleh sistem pakar, salah satunya adalah kaidah produksi.

#### 2.1.6 Teknik Inferensi

Pada sistem pakar berbasis *rule*, domain pengetahuan di presentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data di presentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam database. Jika bagian *IF* (kondisi) dari *rule database* sebagai fakta baru yang ditambahkan, adapun metode yang digunakan adalah *backward chaining* (Sutojo, Mulyanto, & Suhartono, 2011, p. 171).

### **2.1.6.1 Backward Chaining**

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 178) *backward chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada di bagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* di tempatkan di bagian basis data sebagai fakta baru. Jika tidak cocok, simpan premis di bagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal*.

### **2.1.7 Tabel dan Pohon Keputusan**

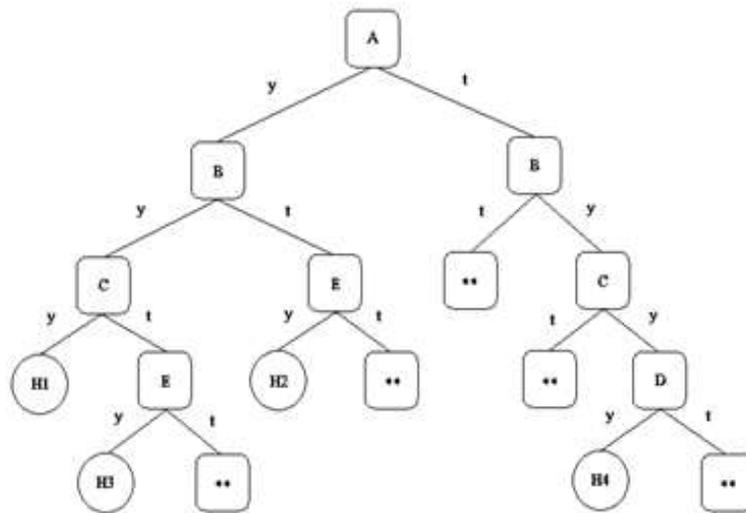
Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang didapatkan dalam domain tertentu yang disajikan pengetahuan dalam bentuk tabel keputusan kemudian dibuat dalam pohon keputusannya. Berikut adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Sri & Sari, 2008).

**Tabel 2.1** Tabel Keputusan

<b>‘Hipotesa</b> <b>Evidence</b>	<b>Hipotesa 1</b>	<b>Hipotesa 2</b>	<b>Hipotesa 3</b>	<b>Hipotesa 4</b>
<i>Evidence A</i>	Ya	Ya	ya	Tidak
<i>Evidence B</i>	Ya	Tidak	ya	Ya
<i>Evidence C</i>	Ya	Tidak	tidak	Ya
<i>Evidence D</i>	Tidak	Tidak	tidak	Ya
<i>Evidence E</i>	Tidak	Ya	ya	Tidak

Sumber: Hartati & Iswanti, (2008)



**Gambar 2.2** Pohon Keputusan  
Sumber : Hartati & Iswanti(2008)

Keterangan:

A = *evidence A*,      H1 = *hipotesa 1*,      y = ya  
 B = *evidence B*,      H2 = *hipotesa 2*,      t = tidak  
 C = *evidence C*,      H3 = *hipotesa 3*, \*\* = *tidak menghasilkan hipotesa tertentu*  
 D = *evidence D*,      H4 = *hipotesa 4*

Dari gambar 2.2 dapat diketahui bahwa *hipotesa* H1 terpenuhi jika memenuhi *evidence* A, B, dan C. *Hipotesa* H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. *Hipotesa* H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. *Hipotesa* H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi node (*evidence*) di atasnya, notasi “t” artinya tidak memenuhi.

## 2.2 Variabel Penelitian

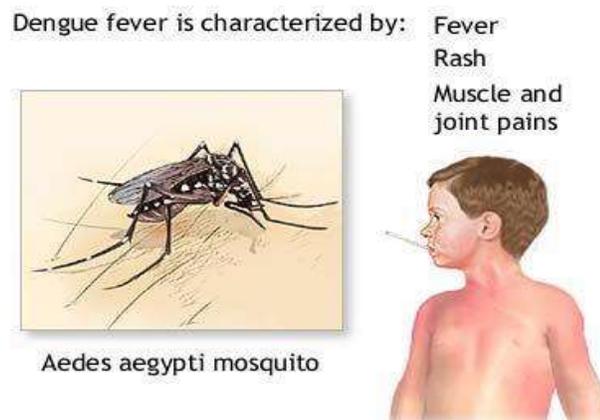
Menurut (Dr.Sudaryono, 2015, p. 17) variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk di pelajari sehingga diperoleh informasi dan kesimpulannya. Peneliti bekerja pada tingkat teoritis dan empiris. Pada tingkat teoritis, perhatiannya tercurah pada pengidentifikasian konsep dan hubungannya dengan proposisi. Pada tahap empiris, pernyataan riset akan di uji pada penelitian ini yang menjadi variabel adalah penyakit pada balita, jenis penyakit pada balita yaitu *Denque haermorrhagic fever*, kejang demam, infeksi saluran pernapasan, demam *thypoid* dan diare.

## 1. Penyakit *Denque haermorhagic fever*

Penyakit *Denque haermorhagic fever* merupakan penyakit yang disebabkan oleh virus denque sejenis virus yang tergolong arbovirus dan masuk kedalam tubuh penderita gigitan nyamuk *Aedes aegypti* (betina).

### a. Tanda dan Gejala

- Suhu tinggi
- Nyeri pada otot seluruh tubuh
- Nyeri dibelakang kepala
- Suara serak
- Batuk
- Epitaksis
- Disuria
- Anoreksia



**Gambar 2.3** *Denque Haermorhagic Fever*  
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Tirah baring
- Diet makan lunak
- Minum banyak
- Pemberian cairan intra vena
- Monitor tanda-tanda vital tiap 3 jam
- Periksa HB, HT, dan trombosit setiap hari
- Pemberian obat anti piretik
- Monitor tanda-tanda pendarahan lebih lanjut
- Pemberian anti biotik
- Bila kejang dapat diberi obat penenang

**2. Penyakit Kejang Demam**

Kejang demam terjadi pada saat suhu meningkat disebabkan oleh suatu proses ekstrakranium. Kejang adalah pembebasan listrik yang tidak terkontrol dari sel syaraf cortex serebral yang ditandai dengan serangan yang tiba-tiba.

a. Tanda dan Gejala

- Ketidakseimbangan kimiawi
- Demam
- Patologis otak
- Ekslampsia
- Idiopatik

- Obat-obatan



**Gambar 2.4** Penyakit Kejang Demam  
Sumber : Data Penelitian (2018)

- b. Solusi dan Penanganannya
- Pemberian Diazepam : dosis awal 0,3-0,5 mg/kg BB/dosis IV 9perlahan, bila kejang belum berhenti dapat diulang dengan dosis ulangan setelah 20 menit
  - Turunkan demam : memberi anti piretik parasetamol atau salisilat 10 mg/kg BB/dosis, kompres air biasa
  - Penanganan suportif : Bebaskan jalan nafas, beri zat asam, jaga keseimbangan cairan dan elektrolit, Pertahankan tekanan darah.

### 3. Penyakit ISPA

Infeksi Saluran Pernapasan atas adalah penyakit infeksi yang menyerang salah satu bagian dan atau lebih dari saluran napas, mulai dari hidung (saluran atas) hingga alveoli (saluran bawah) termasuk jaringan adneksanya, seperti sinus, rongga telinga tengah dan pleura.

a. Tanda dan Gejala

- Rinitis
- Nyeri tenggorokan
- Batuk dengan dahak kuning atau putih kental
- Nyeri retrosternal dan konjungvitis
- Suhu badan meningkat antara 4-7 hari disertai malas
- Mialgia
- Nyeri kepala
- Anoreksia
- Mual
- Muntah dan insomnia



**Gambar 2.5** Penyakit ISPA  
Sumber : Data Penelitian (2018)

b. Solusi dan Penangananya

- Menjaga keadaan gizi agar tetap baik

- Immunisasi
- Menjaga kebersihan perorangan dan lingkungan
- Mencegah anak berhubungan dengan penderita ISPA
- Pemberian obat antibiotik

#### **4. Penyakit Demam Thypoid**

Tifus abdominalis (demam tifoid, enteric fever) ialah penyakit infeksi akut yang biasanya terdapat pada saluran pencernaan dan gangguan kesadaran.

##### **a. Tanda dan Gejala**

- Perasaan tidak enak
- Lesu
- Nyeri kepala
- Pusing dan tidak bersemangat
- Demam : demam berlangsung selama 3 minggu
- Gangguan pada saluran pencernaan
- Gangguan kesadaran
- Panas tinggi yang kontinu
- Kesadaran menurun sampai spoor, koma atau delerium
- Dehidrasi, asidosis, peritonitis, bronkopneumonia
- Gizi yang buruk



**Gambar 2.6** Penyakit Thypoid  
Sumber : Data Penelitian (2018)

- b. Solusi dan Penanganannya
- Isolasikan penderita dan desinfeksi pakaian
  - Lakukan perawatan yang baik untuk menghindari komplikasi, mengingat sakit yang lama, lemah dan anoreksia dan lain-lain
  - Selama masih demam harus istirahat, sampai dengan 2 minggu normal kembali yaitu istirahat total, seminggu kemudian boleh duduk dan selanjutnya boleh berdiri dan berjalan
  - Diet makanan yang mengandung cairan, kalori dan tinggi protein
  - Pemberian antoibiotik dan kloramfenikol

## 5. Penyakit Diare

Diare akut yang awalnya mendadak dan berlangsung singkat dalam beberapa jam sampai 7 atau 14 hari. Proses terjadinya diare akut dipengaruhi dua hal pokok yaitu konsistensi fases dan motilitas usus, umumnya terjadi akibat pengaruh keduanya. Gangguan proses mekanik dan ensimatik, disertai gangguan mukosa, akan mempengaruhi pertukaran air dan elektrolit, sehingga

mempengaruhi konsistensi feses yang terbentuk. Peristaltik saluran cerna yang teratur akan mengakibatkan proses cerna secara ensimatik yang akan mempengaruhi pola defikasi.

a. Tanda dan Gejala

- Mengalami nause
- Muntah
- Nyeri perut sampai kejang perut
- Demam
- Diare
- Kekurangan cairan menyebabkan pasien merasakan haus, lidah kering,tulang pipih menonjol dan turgor kulit menurun
- Suara menjadi serak



**Gambar 2.7** Penyakit Diare  
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Perilaku sehat : Pemberian ASI, Makanan pendamping ASI, Mencuci tangan, menggunakan air bersih yang cukup, menggunakan jamban, membuang tinja yang benar, pemberian imunisasi campak
- Penyehatan lingkungan : penyediaan air bersih, pengolahan sampah, sarana pembuangan air limbah
- Pemberian obat oralit, diberikan cairan ringer laktat, bila tak tersedia diberikan cairan NaCl isotonik ditambah satu ampul Na bikarbonat 7,5 % 50 ml
- Pemberian obat anti diare bersifat simptomatik
- Pemeriksaan darah tepi langka
- Pemeriksaan urine lengkap

### 2.3 *Software* Pendukung

*Software* adalah sebuah perangkat yang berfungsi sebagai pengatur aktivitas kerja komputer dan seluruh intruksi yang mengarah pada sistem komputer. *Software* dibuat dengan menggunakan bahasa pemrograman yang ditulis atau diciptakan oleh *programer* yang selanjutnya di komplikasi dengan aplikasi kompuler sehingga menjadi sebuah kode yang nantinya akan dikenali oleh mesin *hardware*.

### 2.3.1 *Personal Home Page* (PHP)

#### 2.3.1.1 Definisi PHP



**Gambar 2.8** Logo PHP  
Sumber: Data Penelitian (2018)

PHP merupakan suatu bahasa *scripting* khususnya digunakan untuk web *delevopment*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, JavaScript, Jquery, Ajax. Pada umumnya PHP lebih banyak digunakan bersamaan dengan file bertipe HTML. Dengan menggunakan PHP membuat *website powerful* yang dinamis disertai manajemen *database* (Hidayatullah & Kawistara, 2017:224).

#### 2.3.1.2 Kelebihan PHP

PHP berbasis *server side scripting*. PHP dapat melakukan tugas-tugas yang dilakukan dengan mekanisme CGI seperti mengambil, mengumpulkan data dari database, *meng-generate* halaman dinamis, bahkan menerima dan mengirim cookie seperti dibawah ini:

1. PHP Berbasis *Server Side Scripting*
2. *Command Line Scripting*
3. PHP dapat Membuat Aplikasi Dekstop
4. Digunakan untuk Berbagai Macam *Platform Os*
5. Mendukung Berbagai Macam *Web Server*
6. *Object Oriented Programming* atau *Procedural*
7. Mendukung Banyak RDMS
8. Mendukung Banyak Komunikasi
9. Pengolahan Teks yang Sangat Baik
10. Membuat Program PHP

### **2.3.2 XAMPP**

#### **2.3.2.1 Definisi XAMPP**



**Gambar 2.9** Logo *Xampp*  
Sumber : Data Penelitian (2018)

Menurut (Hidayatullah & Kawistara, 2017:123) *Web Server* merupakan tempat dimana menyimpan aplikasi *web*, kemudian mengaksesnya melalui

internet. *Web server* yang sangat terkenal adalah *apache*. *Apache web server* bisa digunakan di windows, linux dan MAC.

### 2.3.2.2 APACHE

Manurut Winarto ada baiknya kita mengenali aplikasi yang dikemas oleh *XAMPP*.

#### 1. *Apache HTTP Server*

*Apache HTTP Server* merupakan aplikasi untuk *server web* terpopuler di dunia, dengan pangsa pasar pada bulan Maret 2009 mencapai 46%, atau dengan kata lain 46% dari seluruh *website* yang ada di *internet* dijalankan menggunakan *Apache*. Dan pada tahun 2009 ini, *Apache HTTP server* tercatat sebagai aplikasi *server web* pertama yang menembus angka penggunaan mencapai 100 juta situs *web* di seluruh dunia.

Pada Skema tersebut, halaman *web* statis yang diminta *client* akan langsung dikirimkan begitu saja oleh *server*, begitu pula jika *client* mengklik sebuah link untuk meminta halaman lain. Namun pada *Apache* yang dipaketkan oleh XAMPP ini, sudah terdapat dua modul pengolah pemrograman di sisi *server* (*server-side scripting*), yaitu PHP dan *Perl*. Hal ini memungkinkan kita memanfaatkan *web server* untuk menginstall beberapa aplikasi berbasis *web* yang sudah disertakan di dalam CD, atau untuk mempelajari pembuatan *website* dinamis menggunakan bahasa pemrograman tersebut di *server* lokal.

Sebagaimana disebutkan sebelumnya, *Apache* memberikan kemampuan sebuah *web server* pada komputer kita, dan PHP memungkinkan kita menjalankan

sebuah *website* dinamis yang menggunakan bahasa pemrograman PHP. Namun aplikasi berbasis *web* yang terdapat pada CD tidak bisa diinstall jika kita belum menyiapkan sebuah *database server* atau *server basis data* yang sesuai. *Database server* dibutuhkan untuk menyediakan mekanisme penyimpanan data secara terstruktur, efektif, dan efisien. MySQL yang dipaketkan dalam XAMPP merupakan aplikasi *server database* yang mumpuni, dan banyak digunakan pada aplikasi berbasis *web*. Bahkan banyak *website* besar dengan trafik yang tinggi memanfaatkan MySQL untuk penyimpanan basis datanya. Sebut saja *Flickr*, *Facebook*, *Wikipedia*, *Google*, *Nokia* dan *YouTube* yang secara resmi telah membeberkan bahwa *website* mereka menggunakan MySQL sebagai *database server*.

### 2.3.2.3 PHP MyAdmin



**Gambar 2.10** Logo *phpMyAdmin*  
Sumber : Data Penelitian (2018)

Salah satu alat paling ciamik untuk mengelola *database MySQL* adalah *PhpMyAdmin*. Memanfaatkan *phpMyAdmin*, proses pengaturan *privilege* dan manipulasi *database MySQL* lebih mudah dilakukan menggunakan antarmuka *web* oleh orang awam sekalipun.

### 2.3.3 MySQL

#### 2.3.3.1 Definisi MySQL



**Gambar 2.11** Logo MySQL

Sumber : Pengolahan Data Penelitian (2017)

MySQL merupakan salah satu DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi *web* seperti *PostgreSQL*, *SQL Server*, *MS Access* dari *Microsoft*, *DB2* dari *IBM*, *Oracle* dan *Oracle Corp*, *Dbase*, *FoxPro*. Kelebihan dari MySQL adalah gratis, handal, selalu di *update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di bundling dengan *web server* sehingga proses instalasinya menjadi lebih mudah(Hidayatullah & Kawistara, 2017, p. 175).

#### 2.3.3.2 Cara Kerja MySQL

Elemen penting yang perlu diketahui dalam mempelajari SQL adalah sebagai berikut:

1. *Data Definition Language* (DDL), yaitu *statemen-statement* yang berhubungan dengan pembuatan objek (misalnya tabel) dan pengelolaan strukturnya.
2. *Data Manipulation Language* (DML), yaitu *statemen-statement* yang berhubungan dengan manipulasi data di dalam tabel.
3. *Data Control Language* (DCL), yaitu *statemen-statement* kontrol seperti *GRANT* dan *REVOKE*.
4. *Transactional Control Language* (TCL), yaitu *statemen* yang digunakan untuk mengatur transaksi data seperti *START TRANSACTION*, *SAVEPOINT*, *COMMIT*, dan *ROLLBACK*.
5. *Data Query Language* (DQL), yaitu *statemen* yang mengacu ke seleksi data seperti *SELECT* dan *SHOW*.

Untuk memanipulasi data pada tabel-tabel yang terdapat di dalam suatu *database*, anda perlu mempelajari perintah-perintah SQL dari MySQL. Berikut ini perintah-perintah yang perlu anda pelajari secara lebih detail .

1. *SELECT*: digunakan untuk mengambil data dari *database*.
2. *DELETE*: digunakan untuk menghapus data dari *database*.
3. *INSERT*: digunakan untuk memasukkan data baru ke dalam *database*.
4. *REPLACE*: digunakan untuk mengganti data di dalam *database*. Jika terdapat *record* yang sama dalam suatu tabel, perintah ini akan menimpa *record* tersebut dengan data baru.
5. *UPDATE*: digunakan untuk mengubah data di dalam tabel.

Perintah-perintah diatas hanya digunakan untuk memanipulasi data. Untuk memanipulasi struktur objek *database*, gunakan perintah-perintah berikut:

1. *CREATE*: digunakan untuk membuat *database*, tabel, atau *indeks*.
2. *ALTER*: digunakan untuk memodifikasi struktur dari suatu tabel.
3. *DROP*: digunakan untuk menghapus *database*, tabel, atau *indeks*.

### 2.3.4 *StarUML*

#### 2.3.4.1 Definisi *StarUML*



**Gambar 2.12** Logo *StarUML*  
Sumber : Pengolahan Data Penelitian (2017)

*StarUML* merupakan salah satu *CASE* (*Computer-Aided Software Engineering*) *tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak

### 2.3.5 *Notepad++*

***Notepad++* adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan**

menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows*. Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, *Notepad++* juga dilisensikan sebagai perangkat *free*. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena *sourceforge.net* sebagai layanan yang memfasilitasi *Notepad++* membebaskannya untuk digunakan. Beberapa daftar bahasa program yang didukung oleh *Notepad++* adalah C, C++, Java, C#, XML, HTML, PHP, Javascript dan masih banyak lagi bahasa program yang didukung. *Notepad++* sangat ringan untuk digunakan, jadi sekalipun komputer yang Anda miliki dengan spesifikasi rendah tetap bisa menggunakannya karena seperti yang Anda ketahui beberapa program untuk menulis kode sekaligus *compiler*-nya biasanya membutuhkan komputer dengan spesifikasi tertentu.

### 2.3.6 *Unified Modelling Language (UML)*

*Unified Modelling Language* atau UML diketahui sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Saat ini UML sudah menjadi bahasa standar dalam penulisan *blueprint software*.

Menurut (A.S & M.Shalahuddin, 2011, p. 113) UML (*unified Modelling Language*) adalah salah standart bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requitment*, membuat analisis, dan desain serta

menggambarkan arsitektur dalam pemrograman berorientasi objek. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan sebuah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

### 1. *Use Case Diagram*

*Use case* diagram diketahui sebagai salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

*Use case* atau diagram *use case* merupakan diagram pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang buat. secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem

informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit atau aktor. Berikut simbol-simbol yang ada pada diagram *use case* (A.S & M.Shalahuddin, 2011).

**Tabel 2.2** *Use case Diagram*

	<i>ACTOR</i> : Sebuah entitas manusia/ mesin yang berinteraksi dengan sistem
	<i>USE CASE</i> : Merepresentasikan fungsionalitas dari suatu sistem / pekerjaan tertentu pada sistem
	ASOSIASI : Menjelaskan hubungan antara <i>use case</i> yang berupa pertukaran informasi
	<i>DEPENDENCY INCLUDE</i> : Menunjukkan suatu <i>use case</i> harus dipenuhi
	<i>DEPENDENCY EXTEND</i> : Menunjukkan suatu <i>use case</i> akan di laksanakan bersifat <i>optional</i>

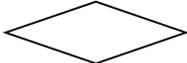
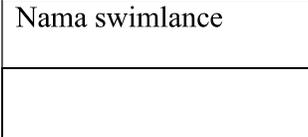
Sumber:(A.S & M.Shalahuddin, 2011)

## 2. *Activity 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 diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (A.S & M.Shalahuddin, 2011).

*Activity diagram* atau diagram aktivitas merupakan suatu jenis diagram khusus dari *statechart diagram* yang mempresentasikan *state-state* dan transisi-transisi yang terjadi pada akhir operasi-operasi. Diagram aktivitas dapat digunakan untuk merepresentasikan *state-state* serta transisi-transisi yang bersifat sinkron (Nugroho,2009).

**Tabel 2.3** *Activity Diagram*

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
	Lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlan 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

**Sumber:** (Rosa A.S dan M.Shalahudin, 2014:161).

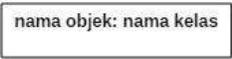
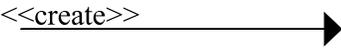
### 3. *Sequence Diagram*

Menurut (A.S & M.Shalahuddin, 2011) *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri tau yang penting semua *use case* yang telah didefenisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut simbol simbol yang ada pada diagram sekuen:

**Tabel 2.4** *Squence Diagram*

Simbol	Deskripsi
<p data-bbox="370 1394 516 1423">Aktor/<i>actor</i></p> 	<p data-bbox="760 1394 1321 1835">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>

<p>Garis hidup/<i>lifeline</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>

Sumber:(A.S & M.Shalahuddin, 2011)

#### 4. *Class Diagram*

Menurut (A.S & M.Shalahuddin, 2011:141–146) Diagram kelas atau *class* diagram menggambarkan struktur segi dari pendefeniann kelas-kelas yang akan dibuat untuk membangun *system*. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik sebaiknya memiliki jenis jenis kelas berikut:

1. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang memiliki tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan pemakai

3. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang harus diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun yang akan disimpan ke basis data.

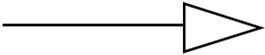
Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi kebasis data, membaca *file teks*, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang didalam kelas perlu memperhatikan apa yang disebut dengan *cohesion dan coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode terkait satu sama lain

sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode yang satu dengan metode yang lain dalam sebuah kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2.5** *Diagram Class*

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

Kebergantungan / <i>dependency</i> .....>	Rlasi antar kelas dengan makna kebergatungan antar kelas
Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna

Sumber:(A.S & M.Shalahuddin, 2011)

#### 2.4. Penelitian terdahulu

1. Nama Penulis : Magdalena Simanjuntak

Judul Jurnal :Sistem pakar diagnosa penyakit diare (gastroentritis) dengan menggunakan forward chaining

Volume / ISSN : Vol .2 No. 3 / 2548-429X

Kesimpulan : Sistem pakar diagnosa penyakit diare ini merupakan suatu sistem dimana masyarakat umum dengan mudah mendiagnosa penyakit diare secara cepat tanpa dibatasi waktu. Sistem pakar diagnosa penyakit diare sangat dibutuhkan oleh masyarakat umum karena dapat mendiagnosa penyakit diare dan mengetahui sebab, solusi, obat dalam penyembuhan diare yang diderita (simanjuntak, 2017).

2. Nama Penulis : Mhd Ridhon, Solikhun dkk

Judul Jurnal : Sistem Pakar diagnosa awal penyakit akibat virus pada anak berbasis *mobile* dengan *Forward chaining*.

Volume / ISSN : 2/ 2540-7600 2540-7597

Kesimpulan : Sistem ini dapat mendiagnosa gejala awal penyakit yang disebabkan virus yang menyerang pada anak-anak khususnya penyakit campak berdasarkan gejala-gejala dan aplikasi sistem pakar ini dapat melakukan diagnosa awal terhadap suatu penyakit yang disebabkan virus serta memberikan informasi mengenai definisi, pengobatan serta pencegahannya, sehingga dapat membantu orangtua dalam mengenali gejala serta jenis-jenis penyakit akibat virus pada anak-anak ( Mhd Ridhon, Solikhun).

3. Nama Penulis : Evi Dewi Sri Mulyani, Irna Nur Restianie

Judul Jurnal : Aplikasi Sistem Pakar untuk mendiagnosa penyakit anak (balita) dengan menggunakan metode *forward chaining*.

Volume / ISSN : 2302-3805

Kesimpulan : Aplikasi sistem pakar ini dapat mendiagnosa suatu penyakit anak (balita) yang sering diderita berdasarkan gejala yang dialami, kemudian dapat memberikan informasi penyakit tersebut beserta solusinya. Dengan diberikan *interface yang user friendly* maka aplikasi sistem pakar ini dapat mudah dipelajari dan mudah digunakan untuk kalangan orang awam, sehingga dapat mempermudah dalam mendiagnosa penyakit (Dewi, Mulyani, & Restianie, 2016).

4. Nama Penulis : Joko Trianto

Judul Jurnal : Penerapan metode *forward chaining* untuk diagnosa penyakit diare pada anak usia 3-5 tahun berbasis *mobile android*.

Volume / ISSN : 2541-1004

Kesimpulan : Tampilan sistem aplikasi yang dibuat masih sederhana, untuk penelitian selanjutnya diharapkan sistem yang dirancang dan dibuat dapat menyempurnakan penelitian sebelumnya agar dapat menghasilkan tampilan yang lebih menarik dan inovatif. Sistem aplikasi yang dibuat belum cukup akurat dalam mendiagnosa penyakit diare, untuk penelitian selanjutnya sistem diharapkan dapat dikembangkan lagi untuk menghasilkan hasil diagnosa yang lebih akurat dibandingkan penelitian sebelumnya. Pembuatan sistem aplikasi ini masih beroperasi secara offline, sebaiknya pengembangan kedepannya dapat beroperasi secara *online* (Trianto, Informatika, Pamulang, & Selatan-indonesia, 2018).

5. Nama Penulis : Bagus fery yanto dkk

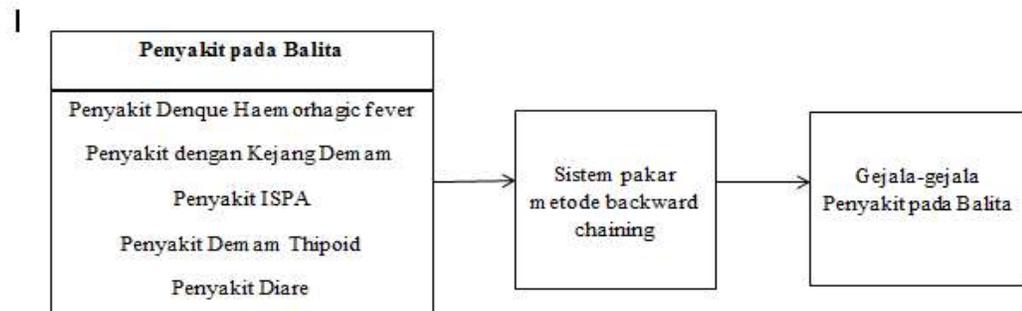
Judul Jurnal : Aplikasi sistem pakar diagnosa penyakit pada anak bawah lima tahun menggunakan metode *forward chaining*.

Volume / ISSN : 3 No.1 / 61-67

**Kesimpulan** : Pembuatan aplikasi sistem pakar diagnosa penyakit pada Balita menggunakan metode *forward chaining* dapat dilakukan dengan tiga tahap. Tahap pertama adalah pengumpulan data dan informasi dari MTBS dan wawancara. Tahap kedua adalah pembuatan *rule* berdasarkan 18 penyakit menggunakan metode *forward chaining*. Tahap ketiga adalah implementasi aplikasi sistem pakar berbasis *android* dengan fitur diagnosa penyakit, riwayat diagnosa dan kumpulan penyakit. Hasil evaluasi menunjukkan tingkat akurasi sistem yang telah dibuat sebesar 82% dengan menggunakan 50 data uji coba (Yanto et al., 2017).

## **2.5 Kerangka Berpikir**

Kerangka pemikiran yang baik akan menjelaskan secara teoritis peraturan antara variabel yang akan diteliti. Serta secara teoritis dapat dijelaskan antara variabel *dependen* dan variabel *independen* serta variabel *intervening* dan *moderating*. (Sugiono, 2014: 60). Berdasarkan dukungan teori dasar dan penelitian terdahulu yang diperoleh, maka dapat disusun kerangka pemikiran sebagai berikut:



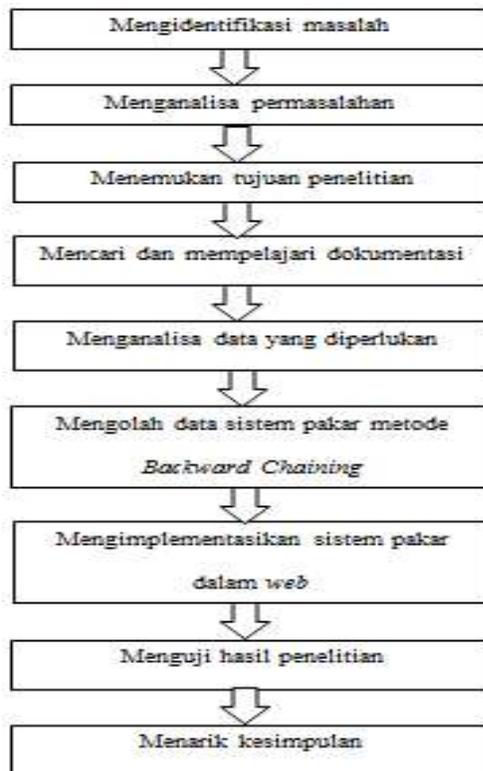
**Gambar 2.13** Kerangka Berpikir  
**Sumber:** Data Penelitian (2018)

Berdasarkan Gambar 2.14 kerangka berpikir input yang pertama adalah jenis-jenis penyakit pada balita kemudian sistem pakar dengan metode *backward chaining*, kemudian jenis-jenis penyakit pada balita yang didapat dari wawancara seorang pakar dan diolah dengan metode *backward chaining* untuk dimasukkan kedalam sebuah *web* kemudian *output* atau keluaran dari penyakit tersebut adalah gejala-gejala dari penyakit pada balita tersebut.

## BAB III METODE PENELITIAN

### 3.1. Desain Penelitian

Desain penelitian yaitu tahapan yang akan dilakukan peneliti untuk mempermudah dalam melakukan penelitian. Desain penelitian sistem pakar diagnosa penyakit pada balita dengan metode *backward chaining* digambarkan seperti pada gambar 3.1 berikut tahapan penelitian yang dilakukan.



**Gambar 3.1** Desain Penelitian  
**Sumber:** Data Penelitian (2018)

Penjelasan desain penelitian pada gambar 3.1 diatas, adalah:

1. Mengidentifikasi Masalah

Penelitian dimulai dengan mengidentifikasi masalah karena masalah merupakan kendala yang harus diselesaikan, agar tercapai tujuan penelitian dengan hasil yang maksimal. Dalam penelitian ini gejala penyakit pada balita ini sangatlah penting permasalahan yang harus dicari solusinya karena masalah dalam penelitian ini yang akan menentukan kualitas suatu penelitian.

2. Menganalisa Permasalahan

Menganalisa permasalahan penyakit dan gejala pada penyakit berdasarkan fakta yang ada.

3. Menemukan Tujuan Penelitian

Setelah menganalisa permasalahan dalam penelitian ini harus menemukan apa tujuan dari penelitian sistem pakar metode *backward chaining* ini.

4. Mencari dan Mempelajari Dokumentasi

Mencari bahan yang akan digunakan pada penelitian ini seperti data gejala, penyakit, data solusi dan cara penanggulangannya, dan mempelajari dari data-data yang sudah ditemukan atau sudah dicari.

5. Menganalisa Data yang Diperlukan

Setelah mencari dan mempelajari dokumentasi dilanjutkan dengan menganalisa data yang diperlukan dalam penelitian seperti penyakit dan gejala dari hasil wawancara seorang pakar dianalisa dulu sebelum masuk ke *design* berikutnya.

#### 6. Mengolah Data Sistem Pakar Metode *Backward Chaining*

Selanjutnya mengolah data sistem pakar dengan metode *backward chaining* yaitu data penyakit dan gejala penyakit pada balita dengan aturan-aturan rule dan pohon keputusan.

#### 7. Mengimplementasi Sistem Pakar dalam *Web*

Selanjutnya mengimplementasi desain sistem yang sudah dibuat menjadi program.

#### 8. Menguji Hasil Penelitian

Pada tahapan ini akan dilihat bagaimana hasil dari pengujian penelitian. Berapa *sampel* yang berhasil dan berapa yang gagal.

#### 9. Menarik Kesimpulan

Setelah itu akan menarik kesimpulan berdasarkan penelitian yang telah dilakukan dan disertakan saran untuk penelitian selanjutnya.

### 3.2 Teknik Pengumpulan Data

Pada penelitian ini informasi dan data yang akurat dapat menunjang proses penelitian. Beberapa metode pengumpulan data, dalam penelitian ini adalah:

#### 1. Studi Kepustakaan

Dengan melakukan studi kepustakaan mengenai kecerdasan buatan, sistem pakar, metode *backward chaining* serta penyakit dan gejala penyakit pada balita melalui jurnal, buku, sumber ilmiah lain.

#### 2. Wawancara

Wawancara langsung dengan pakar tentang permasalahan yang diambil untuk mendapatkan data yang akurat mengenai penyakit dan gejala pada penyakit pada balita. Proses wawancara dilakukan dengan cara melakukan tanya jawab dengan pakar dan pakar memberikan nilai dan jawaban pada setiap gejala-gejala penyakit pada balita tersebut.

### 3.3. Operasional variabel

Variabel yang didapat dalam penelitian ini adalah penyakit pada balita dan indikatornya adalah penyakit-penyakit pada balita yang diambil dari wawancara dengan seorang pakar, dibawah ini akan membahas tentang operasional variabel pada penyakit pada balita.

**Tabel 3.1** Variabel dan Indikator

<b>Variabel</b>	<b>Indikator</b>
Penyakit pada Balita	Denque Haermorhagic Fever
	Kejang Demam
	ISPA
	Demam Thypoid
	Diare

**Sumber:** Data Penelitian (2018)

Pada tabel 3.1 diatas menjelaskan hubungan antara variabel dan indikator. Variabel diatas adalah penyakit pada balita sedangkan indikatornya adalah jenis-jenis

penyakit pada balita yaitu: denque haermorhagic fever, kejang demam, ispa, demam thypoid, diare.

### 3.4 Metode Perancangan Sistem

Dalam perancangan sistem dimana diperlukan suatu keahlian untuk perancangan elemen-elemen komputer yang menggunakan sistem pemilihan peralatan dan perogram untuk sistem yang baru. Dibawah ini akan membahas metode perancangan sistem pakar penyakit pada balita.

#### 3.4.1 Desain Basis Pengetahuan

Sebelum malakukan desain basis pengetahuan, peneliti telah melakukan proses akuisisi pengetahuan dengan mengumpulkan data pengetahuan dan fakta dari sumber-sumber yang tersedia. sumber dan fakta diperoleh dari wawancara dengan pakar sebagai konsultasi penyakit pada balita.

**Tabel 3.2** Data Penyakit pada Balita

Kode	Penyakit
P01	Denque Haermorhagic Fever
P02	Kejang Demam
P03	ISPA
P04	Demam Thypoid
P05	Diare

**Sumber:** Data Penelitian (2018)

Pada tabel 3.2 menjelaskan kode dan jenis penyakit pada balita, kemudian memberi kode penyakit untuk setiap jenis penyakit pada balita.

**Tabel 3.3** Data Gejala Penyakit pada Balita

<b>Kode</b>	<b>Gejala</b>
G01	Suhu tinggi
G02	Nyeri pada otot seluruh tubuh
G03	Nyeri dibelakang kepala
G04	Suara serak
G05	Batuk
G06	Epitaksis
G07	Disuria
G08	Anoreksia
G09	Ketidakseimbangan kimiawi
G10	Demam
G11	Patologis otak
G12	Ekslampsia
G13	Idiopatik

- G14 Obat-obatan
- G15 Rinitis
- G16 Nyeri tenggorokan
- G17 Batuk
- G18 Nyeri Retrosternal dan konjungvitis
- G19 Suhu badan meningkat
- G20 Mialgia
- G21 Nyeri kepala
- G22 Nyeri kepala
- G23 Mual
- G24 Muntah dan insomnia
- G25 Perasaan tidak enak
- G26 Lesuh
- G27 Nyeri kepala
- G28 Pusing dan tidak bersemangat
- G29 Demam
- G30 Gangguan pada saluran pencernaan

- G31 Gangguan kesadaran
- G32 Panas tinggi
- G33 Kesadaran menurun
- G34 Dehidrasi
- G35 Gizi buruk
- G36 Mengalami mause
- G37 Muntah
- G38 Nyeri perut sampai kejang perut
- G39 Demam
- G40 Diare
- G41 Kekurangan cairan
- G42 Suara menjadi serak

**Sumber:** Data Penelitian (2018)

Pada tabel 3.3 menjelaskan gejala-gejala penyakit pada balita kemudian setiap gejala memberikan kode gejala. Data aturan merupakan data yang berisi relasi antara data penyakit dan data gejala penyakit pada balita yang telah diberi kode sebelumnya. Relasi antara data tersebut disusun berdasarkan sumber pengetahuandan fakta yang telah didapatkan.Data aturan ini disusun untuk memudahkan penelitian dalam menyusun kaidah yang akan digunakan sebagai

basis pengetahuan dalam sistem pakar penelitian ini. susunan data aturan yang digunakan dapat dilihat pada tabel dibawah ini.

**Tabel 3.4** Tabel Data Aturan

<b>Kode Penyakit</b>	<b>Kode Gejala</b>
P01	G01,G02,G03,G04,G05,G06,G07,G08
P02	G09,G10,G11,G12,G13,G14
P03	G15,G16,G17,G18,G19,G20,G21,G22,G23,G24
P04	G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
P05	G36,G37,G38,G39,G40,G41,G42

**Sumber:** Data Penelitian (2018)

Berdasarkan tabel 3.4 diatas data aturan yang telah disusun, maka kaidah yang akan digunakan dalam sistem pakar dan tabel keputusannya adalah sebagai berikut:

1. Kaidah 1: *IF* G01 AND G02 AND G03 AND G04 AND G05 AND G06 AND G07 AND G08 *THEN* P01.
2. Kaidah 2: *IF* G09 AND G10 AND G11 AND G12 AND G13 AND G14 *THEN* P02.
3. Kaidah 3: *IF* G14 AND G16 AND G17 AND G18 AND G19 AND G20 AND G21 AND G22 AND G23 AND G24 *THEN* P03.

4. Kaidah 4: *IF* G25 AND G26 AND G27 AND G28 AND G29 AND G30 AND G31 AND G32 AND G33 AND G34 AND G35 *THEN* P04.
5. Kaidah 5: *IF* G36 AND G37 AND G38 AND G39 AND G40 AND G41 AND G42 *THEN* P05.

Berdasarkan kaidah yang telah dibuat maka tabel keputusan adalah sebagai berikut:

**Tabel 3.5** Tabel keputusan gejala dan penyakit pada balita

<b>Bagian</b>		<b>Penyakit</b>				
Kode	Gejala	P01	P02	P03	P04	P05
G01	Suhu tinggi	√				
G02	Nyeri pada otot seluruh tubuh	√				
G03	Nyeri dibelakang kepala	√				
G04	Suara serak	√				
G05	Batuk	√				
G06	Epitaksis	√				
G07	Disuria	√				
G08	Anoreksia	√				
G09	Ketidakseimbangan kimiawi		√			
G10	Demam		√			

G11	Patologis otak	√	
G12	Ekslampsia	√	
G13	Idiopatik	√	
G14	Obat-obatan	√	
G15	Rinitis		√
G16	Nyeri tenggorokan		√
G17	Batuk		√
G18	Nyeri Retrosternal dan konjungvitis		√
G19	Suhu badan meningkat 4-7 hari disertai malas		√
G20	Mialgia		√
G21	Nyeri kepala		√
G22	Anoreksia		√
G23	Mual		√
G24	Muntah dan insomnia		√
G25	Perasaan tidak enak		√

G26	Lesuh	√
G27	Nyeri kepala	√
G28	Pusing dan tidak bersemangat	√
G29	Demam	√
G30	Gangguan pada saluran pencernaan	√
G31	Gangguan kesadaran	√
G32	Panas tinggi yang continue	√
G33	Kesadaran menurun	√
G34	Dehidrasi	√
G35	Gizi buruk	√
G36	Mengalami nausea	√
G37	Muntah	√
G38	Nyeri perut sampai kejang perut	√
G39	Demam	√
G40	Diare	√
G41	Kekurangan cairan	√

G42 Suara menjadi serak ✓

**Sumber:** Data Penelitian (2018)

Pada tabel 3.5 Barisan gejala diberi tanda centang pada kolom penyakit yang memenuhi aturan dari masing masing gejala. Hal ini dibuat untuk mempermudah dalam penyusunan aturan kaidah peroduksi pada sistem pakar yang akan dibuat.

**Tabel 3.6** Tabel Solusi

Solusi gejala dan penyakit pada balita

Kode	P01	P02	P03	P04	P05
S01	✓				
S02	✓				
S03	✓				
S04	✓				
S05	✓				
S06	✓				
S07	✓				
S08	✓				
S09	✓				

S10	√			
S11		√		
S12		√		
S13		√		
S14			√	
S15			√	
S16			√	
S17			√	
S18			√	
S19				√
S20				√
S21				√
S22				√
S23				√
S24				√
S25				√
S26				√

S27	√
S28	√
S29	√

**Sumber:** Data Penelitian (2018)

Pada tabel 3.6 diatas barisan solusi diberi tanda centang pada kolom penyakit yang memenuhi aturan dari masing-masing solusi untuk penyakit .  
Dibawah ini adalah keterangan dari tabel solusi diatas sebagai berikut.

**Keterangan**

S01: Tirah baring

S02: Diet makan lunak

S03: Minum banyak

S04: Pemberian cairan intra vena

S05: Monitor tanda-tanda vital tiap 3 jam

S06: Periksa HB, HT, dan trombosit setiap hari

S07: Pemberian obat anti piretik

S08: Monitor tanda-tanda pendarahan lebih lanjut

S09: Pemberian anti biotik

S10: Bila kejang dapat diberi obat penenang

S11: Pemberian Diazepam : dosis awal 0,3-0,5 mg/kg BB/dosis IV 9perlahan, bila kejang belum berhenti dapat diulang dengan dosis ulangan setelah 20 menit

S12: Turunkan demam : memberi anti piretik parasetamol atau salisilat 10 mg/kg BB/dosis, kompres air biasa

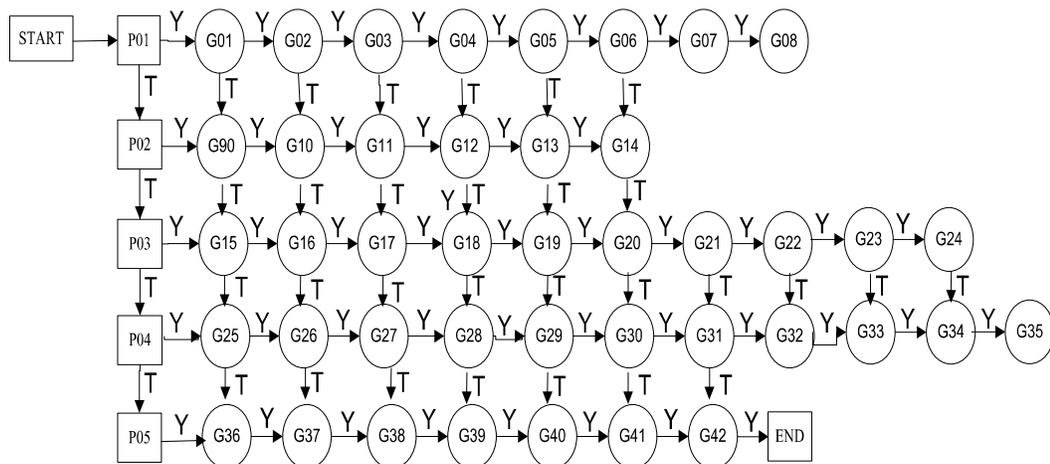
- S13: Penanganan suportif : Bebaskan jalan nafas, beri zat asam, jaga keseimbangan cairan dan elektrolit, Pertahankan tekanan darah.
- S14: Menjaga keadaan gizi agar tetap baik
- S15: Immunisasi
- S16: Menjaga kebersihan perorangan dan lingkungan
- S17: Mencegah anak berhubungan dengan penderita ISPA
- S18: Pemberian obat antibiotik
- S19: Isolasikan penderita dan desinfeksi pakaian
- S20: Lakukan perawatan yang baik untuk menghindari komplikasi, mengingat sakit yang lama, lemah dan anoreksia dan lain-lain
- S22: Selama masih demam harus istirahat, sampai dengan 2 minggu normal kembali yaitu istirahat total, seminggu kemudian boleh duduk dan selanjutnya boleh berdiri dan berjalan
- S22: Diet makanan yang mengandung cairan, kalori dan tinggi protein
- S23: Pemberian antibiotik dan kloramfenikol
- S24: Perilaku sehat : Pemberian ASI, Makanan pendamping ASI, Mencuci tangan, menggunakan air bersih yang cukup, menggunakan jamban, membuang tinja yang benar, pemberian immunisasi campak
- S25: Penyehatan lingkungan : penyediaan air bersih, pengolahan sampah, sarana pembuangan air limbah
- S26: Pemberian obat oralit, diberikan cairan ringer laktat, bila tak tersedia di berikan cairan NaCl isotonik ditambah satu ampul Na bikarbonat 7,5 % 50 ml

S27: Pemberian obat anti diare bersifat simtomatik

S28: Pemeriksaan darah tepi langka

S29: Pemeriksaan urine lengkap

Berdasarkan tabel keputusan pada tabel 3.5 maka pohon keputusannya adalah sebagai berikut



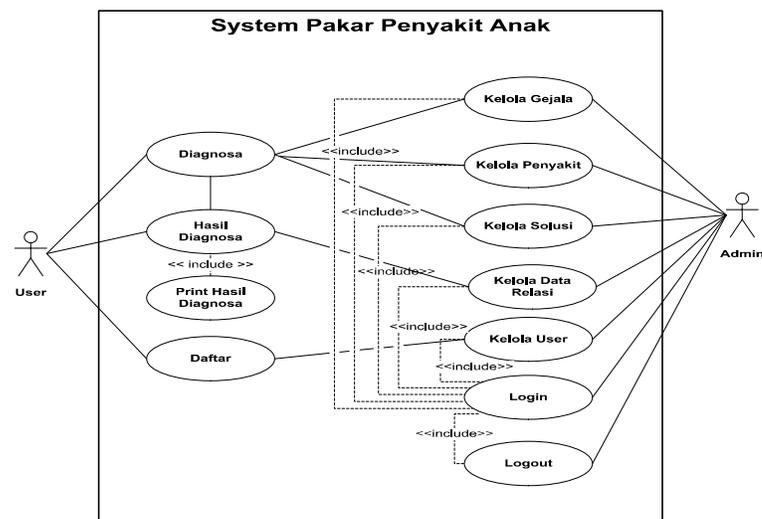
**Gambar 3.2** Pohon Keputusan  
**Sumber:** Data Penelitian (2018)

### 3.4.2 Desain UML (*Unified Modeling Language*)

Menurut (A.S & Shalahuddin, 2013, P. 137, 2013) UML (*Unified Modeling Language*) Merupakan sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Desain UML dibuat untuk memudahkan dalam pembuatan program. Pemodelan UML menggunakan alat bantu *software star UML* versi 2.8.0. Berikut ini adalah UML yang digunakan dalam perancangan program.

## 1. Use Case Diagram Sistem Pakar

Use Case diagram yang akan digunakan pada sistem diagnosis penyakit pada balita seperti pada gambar 3.3 dibawah ini.



**Gambar 3.3** Use Case Diagram  
Sumber: Data Penelitian (2018)

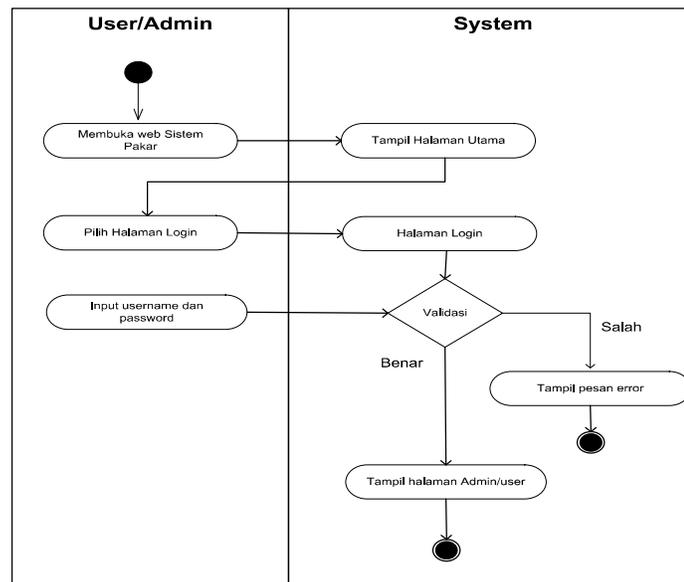
Terdapat 2 aktor utama yaitu *user* dan *admin*. *User* berinteraksi dengan sistem yaitu *register*, *login*, *daftar*, *diagnosa*, *hasil diagnosa*, *print* hasil diagnosa sedangkan *admin* dapat berinteraksi dengan olah data *login*, *kelola gejala*, *kelola penyakit*, *kelola solusi*, *kelola data relasi*, *kelola user*, *logout*.

## 2. Activity Diagram Sistem Pakar

Berikut ini adalah *activity* diagram yang dirancang pada penelitian sistem pakar ini:

### a. Activity Diagram Login Admin dan User

*Activity diagram login admin dan user* merupakan *UML* yang menggambarkan kegiatan pengguna pada halaman khusus *admin* dan *user*.

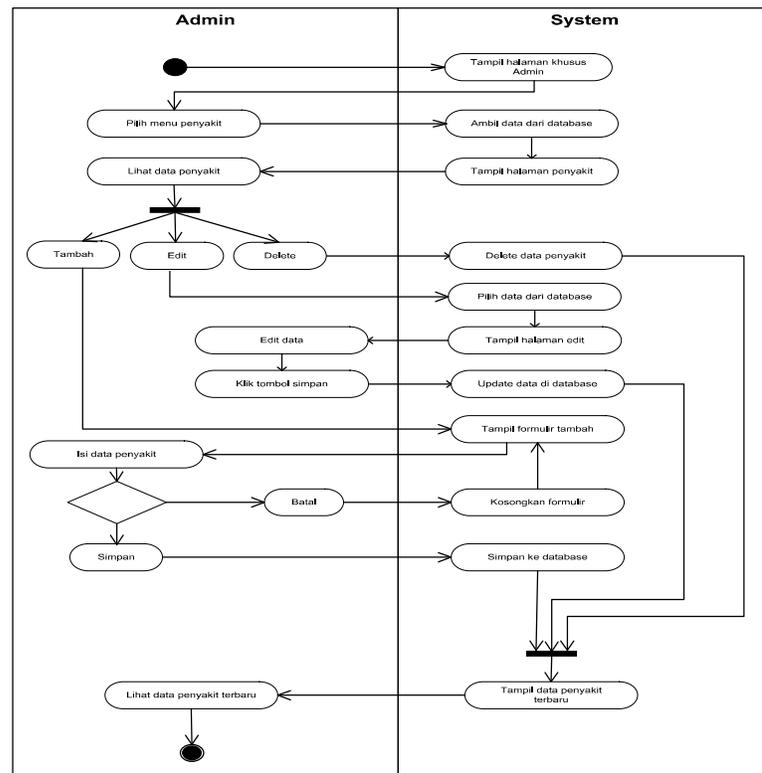


**Gambar 3.4** *Activity Diagram Login Admin dan User*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.4 diatas, tampilan *login admin dan user* membuka *web* sistem pakar kemudian sistem menampilkan halaman *login*, kemudian *user* memasukan *username* dan *password*, jika *username* dan *password* benar akan menampilkan halaman utama, jika *username* dan *password* salah maka sistem akan menampilkan tampilan pesan *error*.

b. *Activity Diagram* Data Penyakit

*Activity diagram* data penyakit merupakan *admin* dalam mengelola data data jenis penyakit.

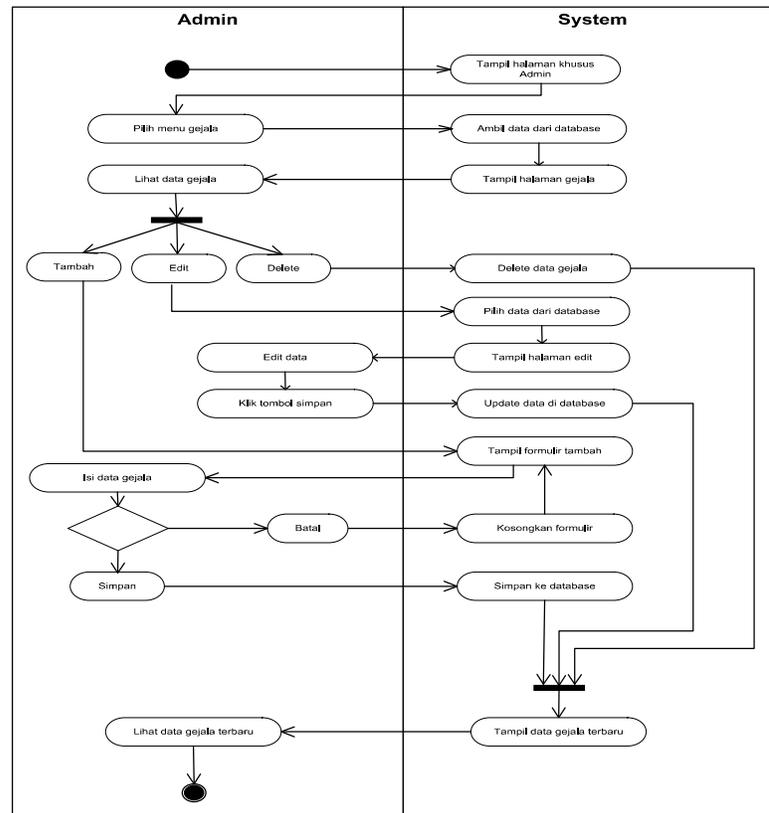


**Gambar 3.5** Activity Diagram Data Penyakit  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.5 diatas, merupakan kegiatan *admin* membuka halaman *web* sistem *admin*, sistem menampilkan halaman menu penyakit, kemudian akan tampil halaman penyakit, lalu *admin* memilih data penyakit terdapat pilihan menu tambah penyakit, kemudian akan muncul tampilan formulir tambah, lalu isi data penyakit, jika batal kosongkan formulir, maka akan kembali ke tampilan formulir, jika simpan maka data akan menyimpan ke *database*, maka akan muncul tampilan data penyakit terbaru, kemudian *admin* pilih menu *edit* akan tampil *edit* data, lalu *klik* tombol simpan akan muncul *update* data di *database*, maka akan muncul tampilan data penyakit terbaru, kemudian *admin* pilih menu *delete* untuk menghapus data penyakit, maka akan muncul tampilan data penyakit terbaru.

c. *Activity Diagram Data Gejala*

*Activity diagram data gejala* merupakan kegiatan *admin* dalam mengelola data gejala penyakit. Berikut adalah *activity diagram data gejala* penyakit dibawah ini.



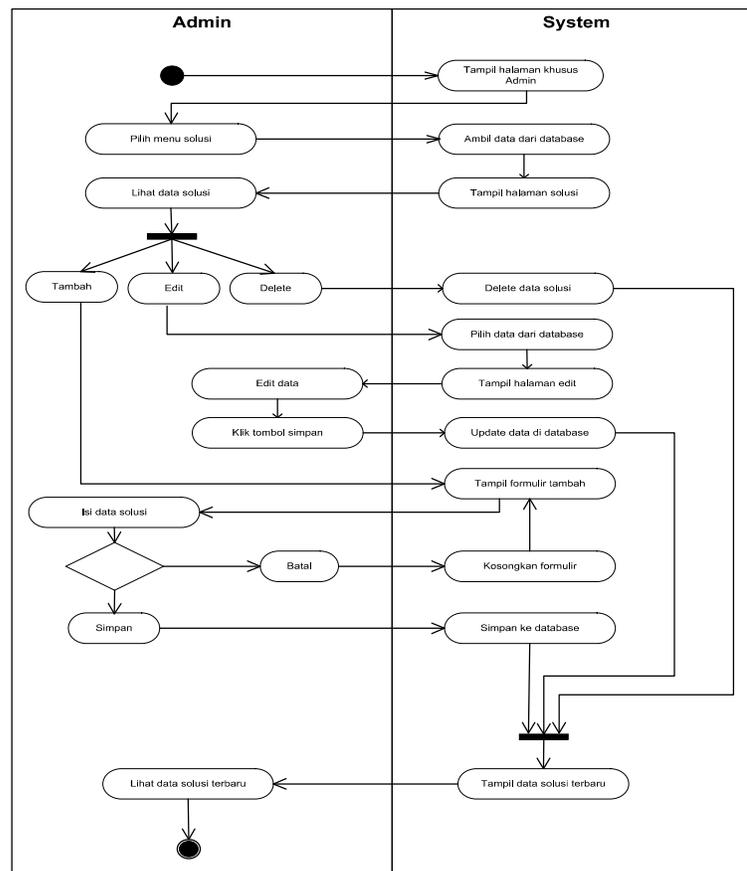
**Gambar 3.6.** *Activity Diagram Data Gejala*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.6 diatas, merupakan kegiatan *admin* membuka halaman *web* sistem *admin*, sistem menampilkan halaman menu gejala, kemudian akan tampil halaman gejala, lalu *admin* memilih data gejala terdapat pilihan menu tambah gejala, kemudian akan muncul tampilan formulir tambah, lalu isi data gejala, jika

batal kosongkan formulir, maka akan kembali ke tampilan formulir, jika simpan maka data akan menyimpan ke *database*, maka akan muncul tampilan data gejala penyakit terbaru, kemudian *admin* pilih menu *edit* akan tampil *edit* data, lalu *klik* tombol simpan akan muncul *update* data di *database*, maka akan muncul tampilan data gejala penyakit terbaru, kemudian *admin* pilih menu *delete* untuk menghapus data gejala, maka akan muncul tampilan data gejala penyakit terbaru.

d. *Activity Diagram Menu Solusi*

*Activity diagram* menu solusi merupakan kegiatan admin dalam mengelola menu solusi penyakit. Berikut adalah *activity diagram* menu solusi dibawah ini

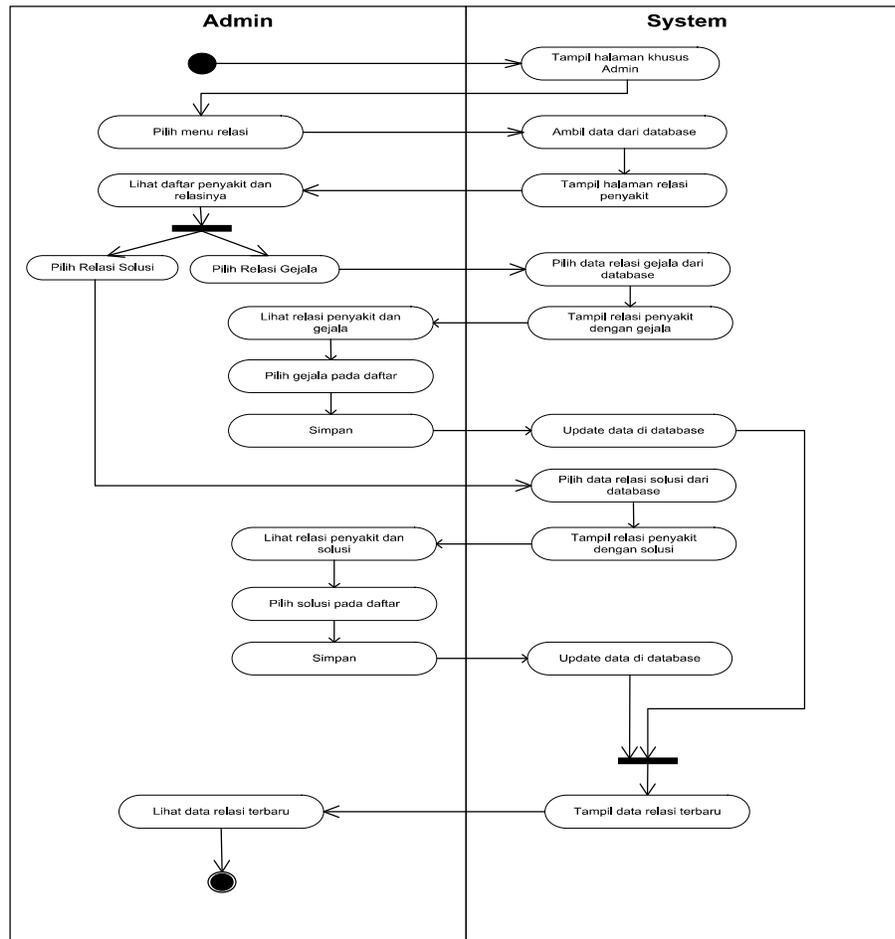


**Gambar 3.7** *Activity Diagram* Solusi**Sumber:** Data Penelitian (2018)

Pada gambar 3.7 diatas, merupakan kegiatan *admin* membuka halaman *web* sistem *admin*, sistem menampilkan halaman menu solusi, kemudian akan tampil halaman solusi, lalu *admin* memilih data solusi terdapat pilihan menu tambah solusi, kemudian akan muncul tampilan formulir tambah, lalu isi data solusi, jika batal kosongkan formulir, maka akan kembali ke tampilan formulir, jika simpan maka data akan menyimpan ke *database*, maka akan muncul tampilan data solusi penyakit terbaru, kemudian *admin* pilih menu *edit* akan tampil *edit* data, lalu *klik* tombol simpan akan muncul *update* data di *database*, maka akan muncul tampilan data solusi penyakit terbaru, kemudian *admin* pilih menu *delete* untuk menghapus data solusi, maka akan muncul tampilan data solusi penyakit terbaru.

e. *Activity Diagram* Menu Relasi

*Activity diagram* menu relasi merupakan kegiatan *admin* dalam mengelola menu relasi penyakit. Berikut adalah *activity diagram* menu relasi dibawah ini.



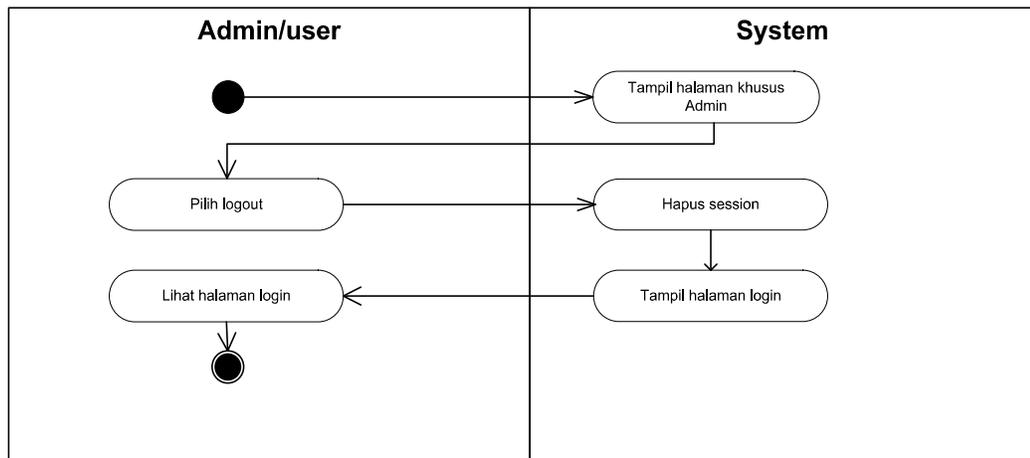
**Gambar 3.8** *Activity Diagram* Menu Relasi

**Sumber:** Data Penelitian (2018)

Pada gambar 3.8 diatas, merupakan kegiatan *admin* membuka halaman *web* sistem *admin*, sistem menampilkan halaman menu relasi penyakit, kemudian akan tampil halaman daftar relasi penyakit, terdapat 2 pilihan relasi penyakit yaitu relasi solusi dan gejala, jika *admin* memilih relasi solusi, maka akan tampil daftar penyakit dan solusi, lalu simpan data solusi, kemudian tampil data relasi terbaru sedangkan relasi gejala, maka akan tampil daftar relasi gejala, lalu pilih relasi gejala, kemudian klik simpan ke *database*, kemudian tampil relasi terbaru.

f. *Activity Diagram Logout*

*Activity diagram logout* merupakan kegiatan *admin* dalam mengelola menu *logout*. Berikut adalah *activity diagram* menu *logout* dibawah ini.

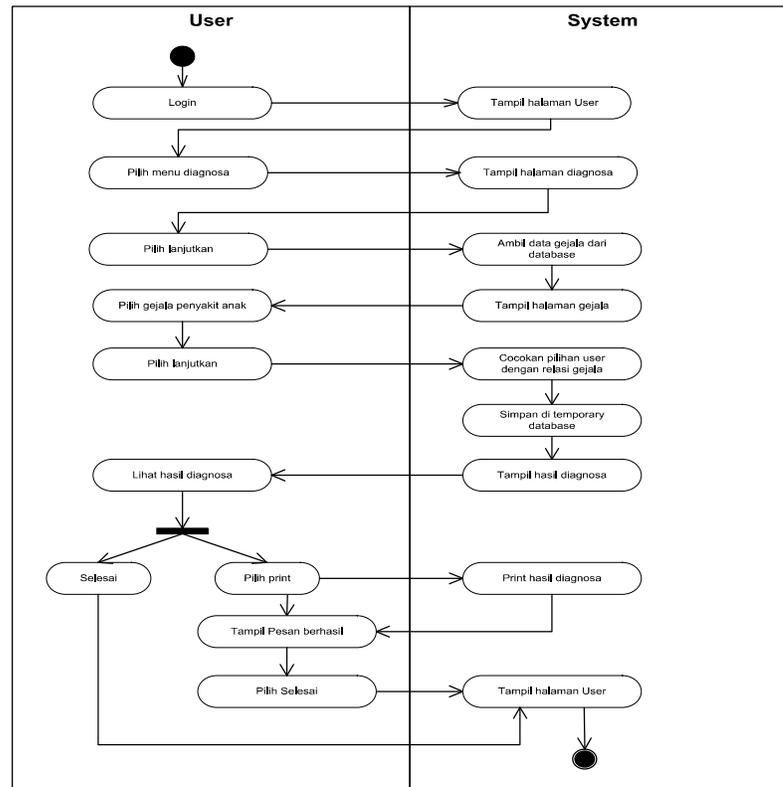


**Gambar 3.9** *Activity Diagram* Menu *Logout*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.9 diatas merupakan kegiatan *admin* membuka halaman *web* sistem *admin*, sistem menampilkan halaman menu *logout*, lalu muncul hapus *session*, maka akan kembali ke halaman *login*.

g. *Activity Diagram* Tampilan *User*

*Activity diagram* tampilan *user* merupakan kegiatan *user* dalam mengelola tampilan *user*/pengguna . Berikut adalah *activity diagram* menu *diagram* tampilan *user* dibawah ini.

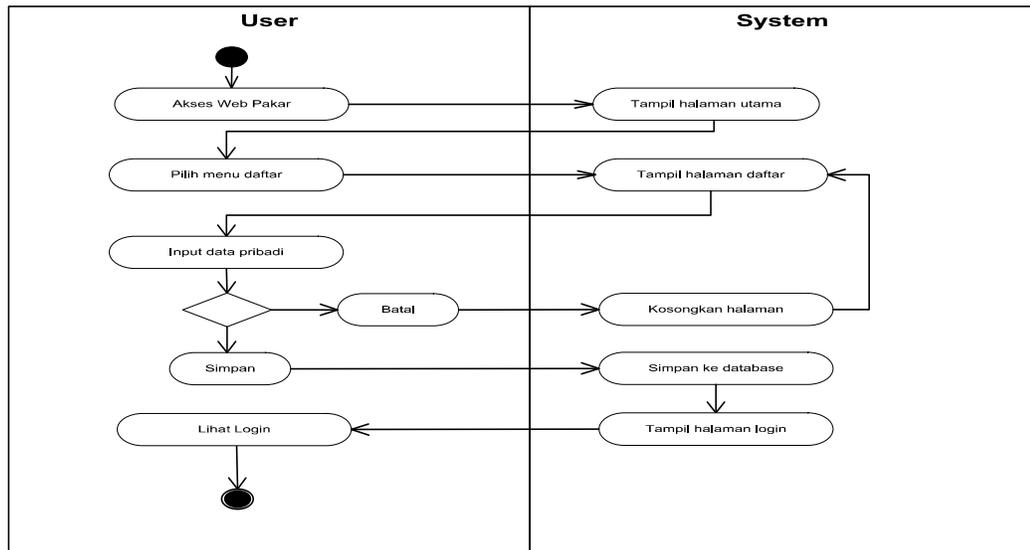


**Gambar 3.10** *Activity Diagram Tampilan User*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.10 diatas, *user* mulai membuka *web* sistem pakar kemudian sistem menampilkan halaman *login* , kemudian *user* memilih menu diagnosa maka akan tampil halaman diagnosa, selanjutnya pilih gejala penyakit anak, kemudian cocokkan pilihan *user* dengan relasi gejala, lalu simpan di temporary database maka akan muncul hasil diagnosa, jika pilih *print* maka tampil hasil *print* pesan berhasil pilih selesai maka akan tampil halaman *user*.

#### h. *Activity Diagram Menu Daftar*

*Activity diagram* menu daftar merupakan kegiatan *user* dalam memilih daftar pada *web* sistem pakar . Berikut adalah *activity diagram* menu daftar *user* dibawah ini.

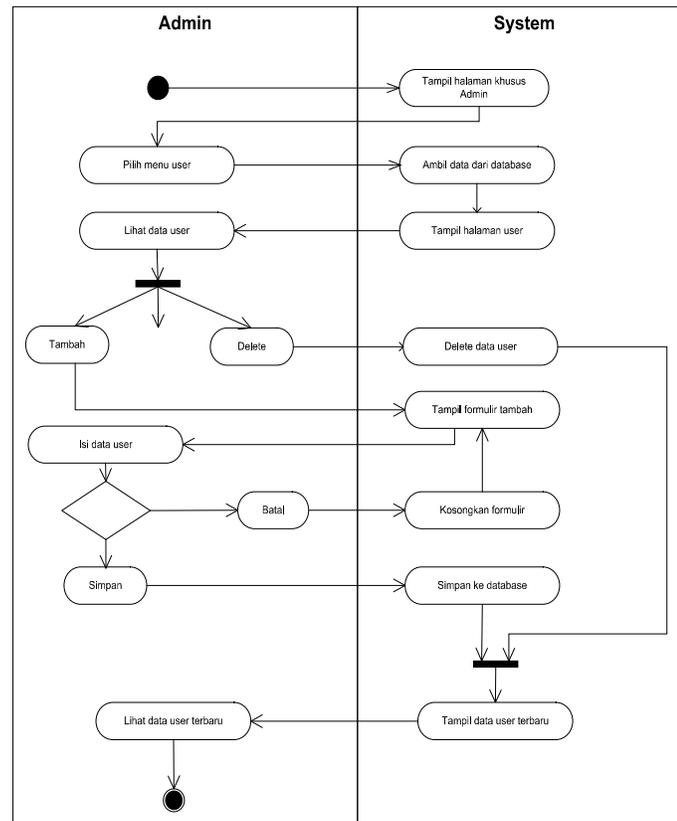


**Gambar 3.11** *Activity Diagram* Daftar  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.11 diatas, *user* membuka *web* sistem kemudian sistem tampil halaman utama depan, kemudian pilih daftar maka akan tampil halaman daftar, sistem akan memberi perintah input data pribadi, kemudian *user* dapat mengklik simpan apabila ingin menyimpan data, dan sistem akan menampilkan batal apabila *user* ingin membatalkan.

i. *Activity Diagram* Menu Kelola *User*

*Activity diagram* menu kelola *user* merupakan kegiatan *user* dalam mengelola admin pada *web* sistem pakar . Berikut adalah *activity diagram* menu kelola *user* dibawah ini.

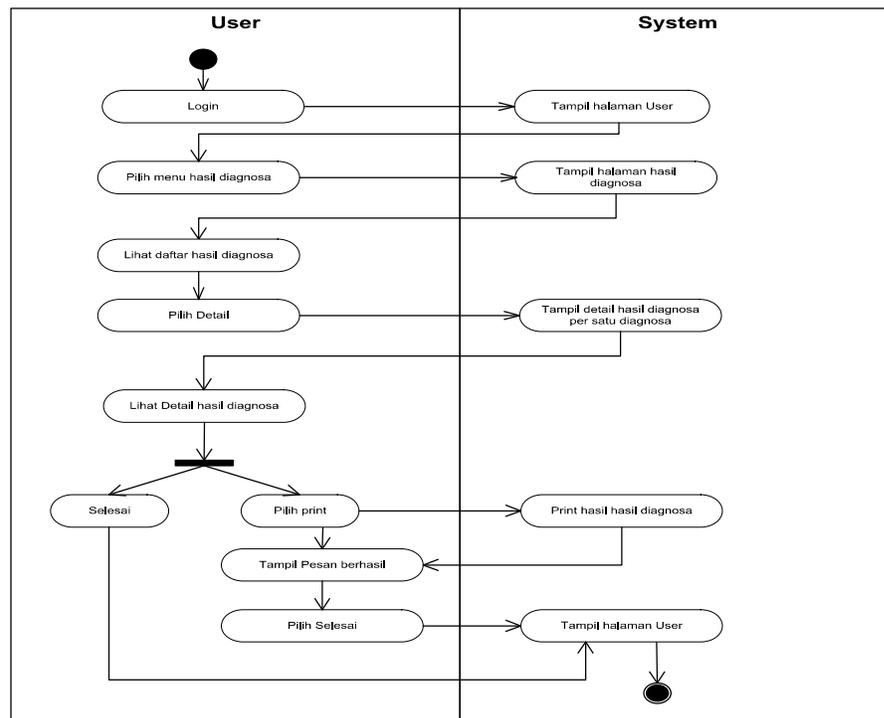


**Gambar 3.12** *Activity Diagram Kelola User*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.12 diatas, *user* membuka *web* sistem kemudian sistem tampil halaman menu utama, kemudian pilih menu *user* maka akan tampil halaman user, sistem akan memberi perintah lihat data *user*, kemudian *user* dapat mengklik tambah apabila ingin menambah data user, dan sistem akan menampilkan *delete* apabila *user* ingin menghapus data *user*.

j. *Activity Diagram Menu Hasil Diagnosa*

*Activity diagram* menu hasil diagnosa merupakan kegiatan *user* dalam hasil diagnosa penyakit . Berikut adalah *activity diagram* menu hasil diagnosa dibawah ini.



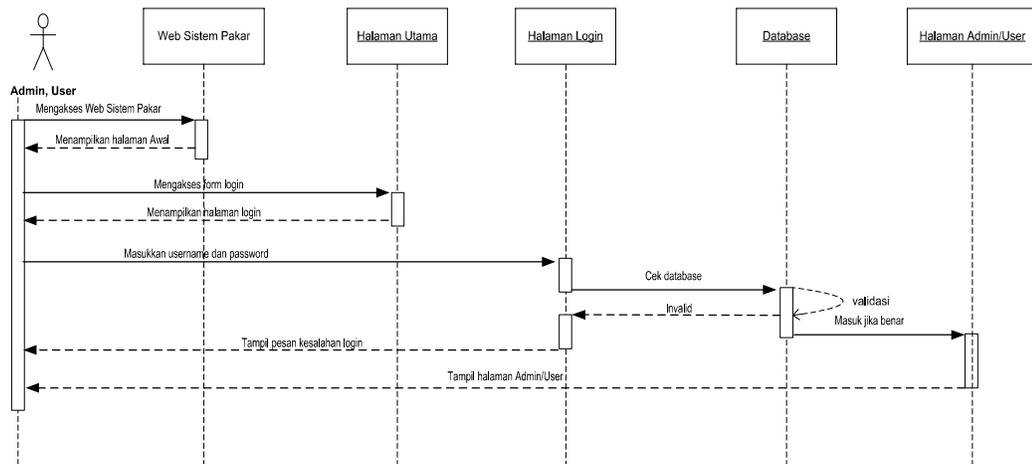
**Gambar 3.13** *Activity Diagram Hasil Diagnosa*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.13 diatas, *user* membuka *web* sistem kemudian sistem tampil halaman *login*, kemudian pilih menu *user* maka akan tampil halaman *user*, lalu pilih menu diagnosa maka akan tampil halaman hasil diagnosa, kemudian lihat daftar hasil diagnosa sistem akan memberi perintah lihat detail hasil diagnosa, jika pilih print lalu tampil hasil print diagnosa, selesai kembali ke halaman *user*.

### 3. *Sequence Diagram Sistem Pakar*

*Sequence Diagram* Menggambarkan kelakuan obyek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antara objek. (A.S \$ Shalahuddin,, 2011,P.137)

#### a. *Sequence Diagram Login Admin dan User*

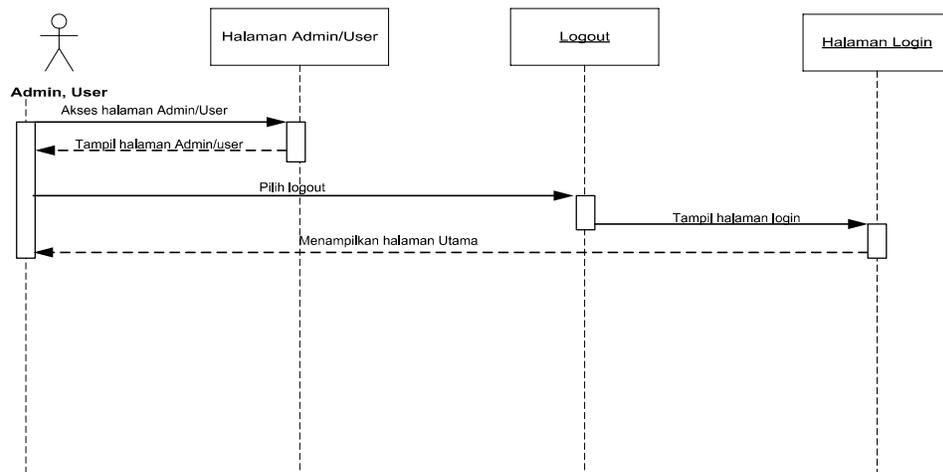


**Gambar 3.14** *Sequence Diagram Login Admin dan User*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.14 diatas, *admin* dan *user* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman awal *admin* dan *user*. Kemudian mengakses *form login*, maka *web* sistem pakar tampil halaman *login*, *admin* dan *user* memasukkan *username* dan *password* pada *form login*. Dari halaman *login* *admin* dan *user* sistem akan mengecek *username* dan *password* yang sudah dimasukan kemudian sistem akan terhubung dengan *database*, setelah itu *username* dan *password* akan dicocokkan oleh sistem, jika *username* dan *password* benar maka halaman utama *admin* dan *user* akan ditampilkan. Jika salah *username* dan *password* maka sistem akan menampilkan pesan kesalahan dihalaman menu *login*.

b. *Sequence Diagram Logout Admin dan User*

*Sequence diagram* tampilan *user* merupakan kegiatan *admin* dalam mengelola semua pengguna. Berikut adalah *sequence diagram* dibawah ini.

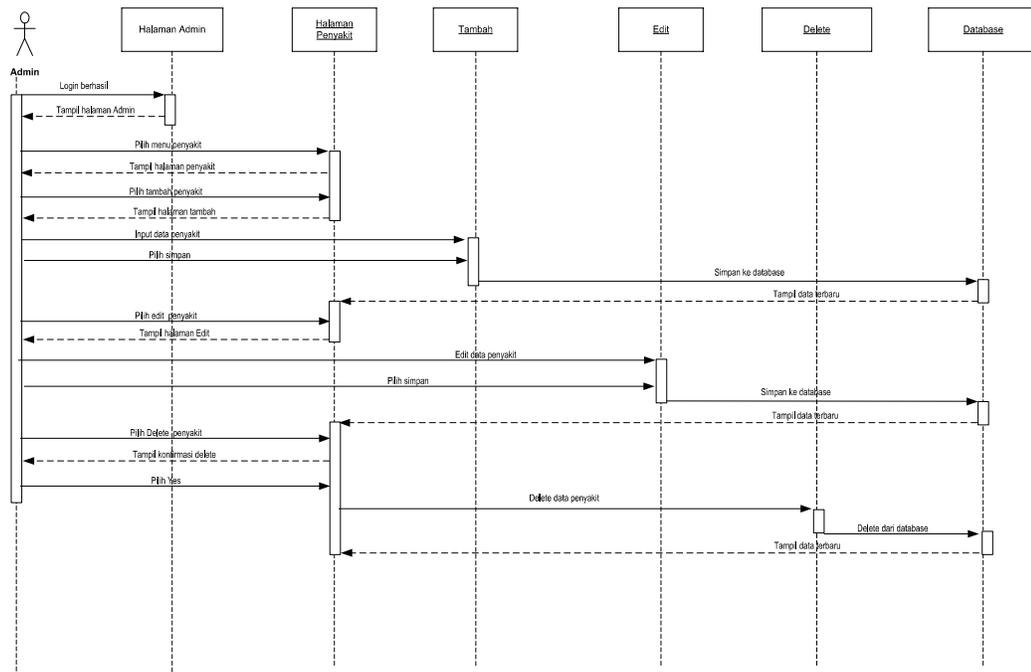


**Gambar 3.15** *Sequence Diagram Logout Admin dan User*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.15 diatas, *admin* dan *user* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman awal *admin* dan *user*. Kemudian *admin* dan *user* pilih menu *logout*, maka sistem akan kembali ke halaman *login* dan *web* sistem akan menampilkan ke halaman utama.

c. *Sequence Diagram Penyakit*

*Sequence diagram* penyakit merupakan kegiatan *admin* dalam mengelola semua pengguna. Berikut adalah *sequence diagram* dibawah ini.

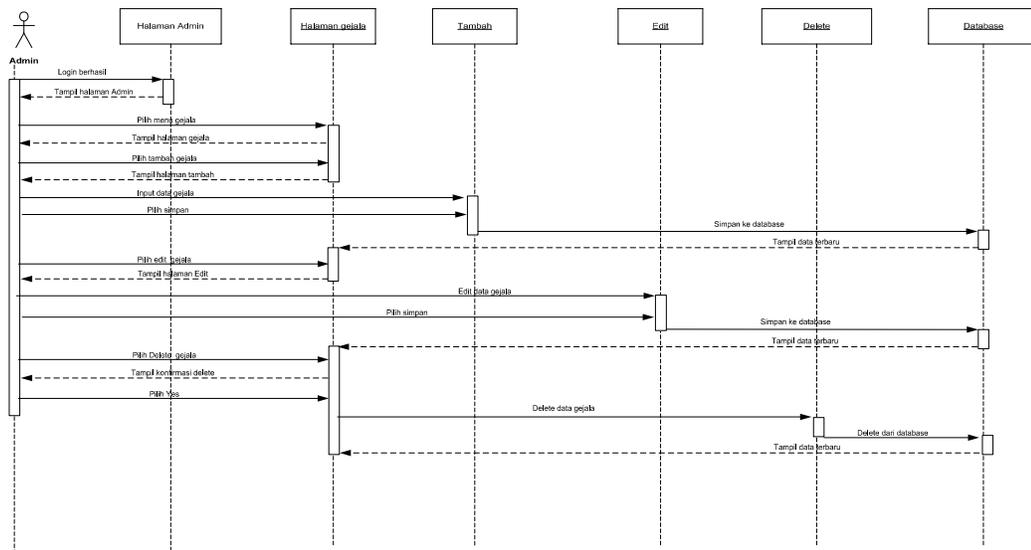


**Gambar 3.16** *Sequence Diagram Penyakit*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.16 diatas *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu penyakit, sistem akan menampilkan halaman penyakit, *admin* pilih tambah penyakit, kemudian sistem akan menampilkan halaman tambah penyakit, lalu *admin* pilih input penyakit, kemudian sistem akan tampilkan halaman input penyakit dan pilih simpan, maka akan tersimpan ke *database*, dan data yang tersimpan akan tampil data baru. Selanjutnya *admin* pilih *edit* penyakit, kemudian tampil halaman edit penyakit. *Admin* melakukan edit data penyakit dan pilih simpan, maka data yang sudah di *edit* akan tersimpan ke *database*, lalu tampil data terbaru. Selanjutnya *admin* pilih *delete* penyakit. Setelah *admin delete* data, maka data akan terhapus dari *database*, kemudian sistem akan menampilkan data terbaru.

d. *Sequence Diagram* Kelola Gejala

*Sequence diagram* kelola gejala merupakan kegiatan *admin* dalam mengelola semua pengguna. Berikut adalah *sequence diagram* dibawah ini.



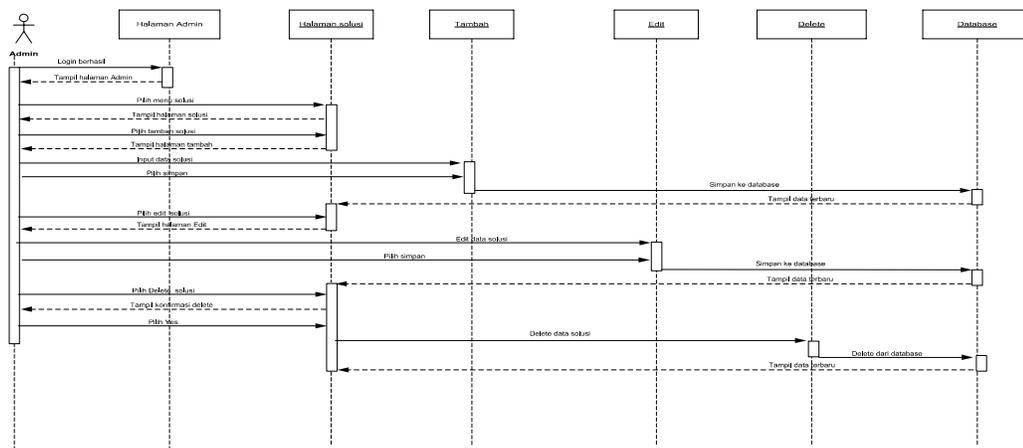
**Gambar 3.17** *Sequence Diagram* Kelola Gejala  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.17 diatas, *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu gejala, sistem akan menampilkan halaman gejala, *admin* pilih tambah gejala, kemudian sistem akan menampilkan halaman tambah gejala, lalu *admin* pilih input gejala, kemudian sistem akan tampilkan halaman input gejala dan pilih simpan, maka akan tersimpan ke *database*, dan data yang tersimpan akan tampil data baru. Selanjutnya *admin* pilih *edit* gejala, kemudian tampil halaman edit gejala. *Admin* melakukan edit data gejala dan pilih simpan, maka data yang sudah di *edit* akan tersimpan ke *database*, lalu tampil data terbaru. Selanjutnya *admin*

pilih *delete* gejala. Setelah *admin delete* data, maka data akan terhapus dari *database*, kemudian sistem akan menampilkan data terbaru.

e. *Sequence Diagram* Solusi Penyakit

*Sequence diagram* solusi penyakit merupakan kegiatan *admin* dalam mengelola solusi penyakit semua pengguna. Berikut adalah *sequence diagram* dibawah ini.



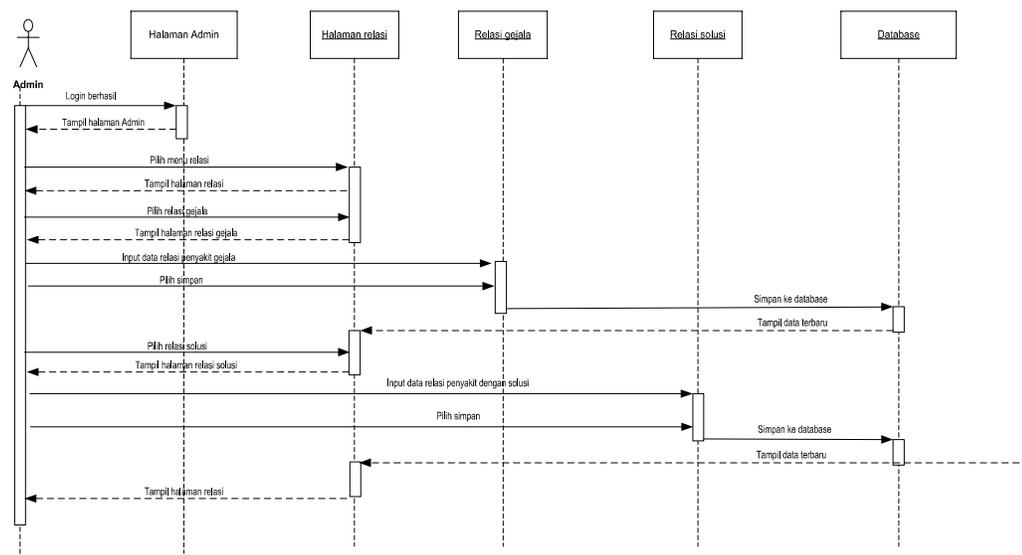
**Gambar 3.18** *Sequence Diagram* Solusi  
Sumber: Data Penelitian (2018)

Pada gambar 3.18 diatas, *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu kelola solusi penyakit, sistem akan menampilkan halaman solusi penyakit, *admin* pilih tambah solusi penyakit, kemudian sistem akan menampilkan halaman tambah solusi penyakit, lalu *admin* pilih input solusi penyakit, kemudian sistem akan tampilkan halaman input solusi penyakit dan pilih simpan, maka akan tersimpan ke *database*, dan data yang tersimpan akan tampil data baru. Selanjutnya *admin* pilih *edit* solusi penyakit, kemudian tampil halaman edit solusi penyakit. *Admin* melakukan edit data solusi penyakit dan pilih

simpan, maka data yang sudah di *edit* akan tersimpan ke *database*, lalu tampil data terbaru. Selanjutnya *admin* pilih *delete* solusi penyakit. Setelah *admin delete* data, maka data akan terhapus dari *database*, kemudian sistem akan menampilkan data terbaru.

f. *Sequence Diagram* Kelola Relasi

*Sequence diagram* kelola relasi merupakan kegiatan *admin* dalam mengelola relasi semua pengguna. Berikut adalah *sequence diagram* dibawah ini.



**Gambar 3.19** *Sequence Diagram* Relasi

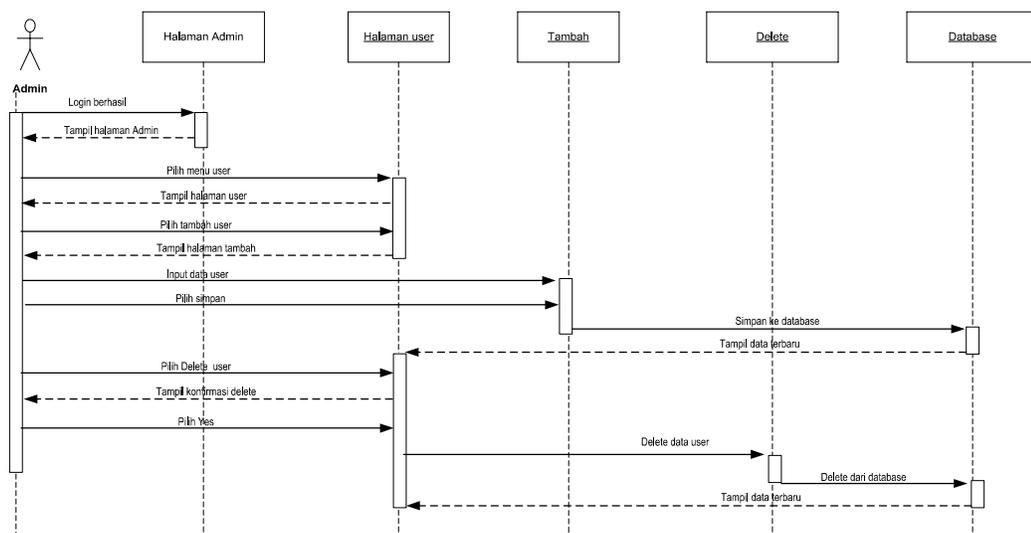
**Sumber:** Data Penelitian (2018)

Pada gambar 3.19 diatas, *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu relasi, sistem akan menampilkan halaman menu relasi, kemudian *admin* pilih relasi gejala, sistem akan menampilkan halaman relasi gejala, lalu *admin* pilih input data relasi penyakit gejala dan pilih simpan, maka akan tersimpan ke *database*, dan data yang tersimpan akan tampil data baru.

Selanjutnya *admin* pilih relasi solusi, kemudian tampil halaman relasi solusi . Setelah itu *admin* melakukan input data relasi penyakit dengan solusi dan pilih simpan, maka data akan tersimpan ke *database*, kemudian sistem akan menampilkan data terbaru, kemudian tampil halaman relasi.

g. *Sequence Diagram Kelola User*

*Sequence diagram* kelola *user* merupakan kegiatan *admin* dalam mengelola semua *user/pengguna*. Berikut adalah *sequence diagram* dibawah ini.



**Gambar 3.20** *Sequence Diagram Kelola User*

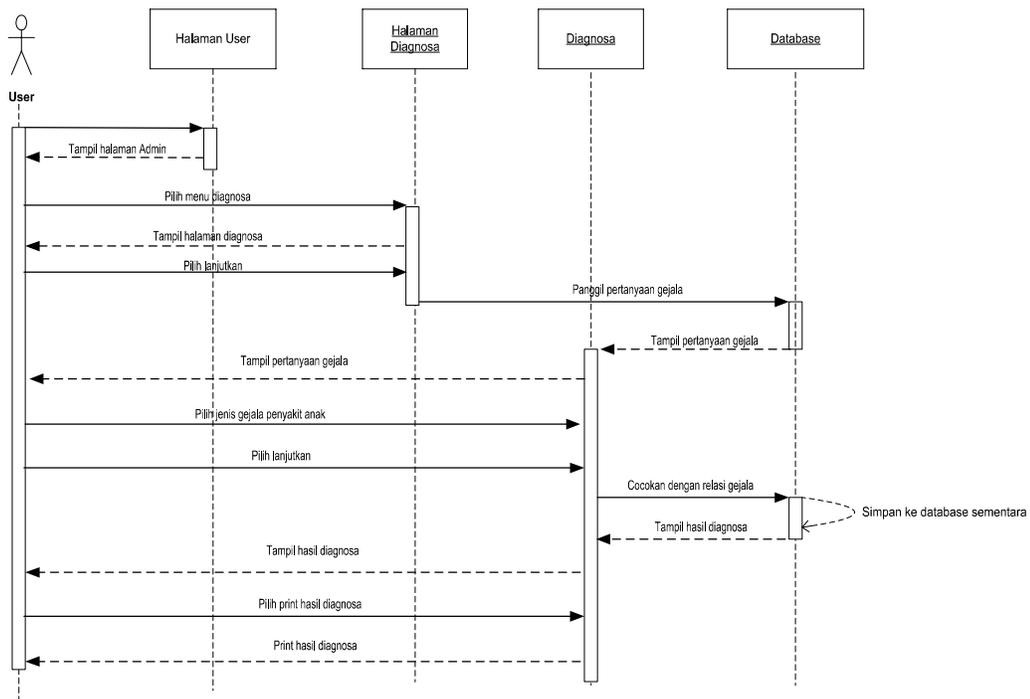
**Sumber:** Data Penelitian (2018)

Pada gambar 3.20 diatas *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu *user*, sistem akan menampilkan halaman *user*, *admin* pilih tambah *user*, kemudian sistem akan menampilkan halaman tambah *user*, lalu *admin* pilih input data *user*, kemudian sistem akan tampilkan halaman input *user* dan pilih simpan, maka akan tersimpan ke *database*, dan data yang tersimpan akan tampil data baru. Selanjutnya *admin* pilih *delete user*, kemudian tampil halaman *delete user*, setelah

itu tampil konfirmasi *delete user*, maka data akan terhapus dari *database*, kemudian sistem akan menampilkan data terbaru.

#### h. *Sequence Diagram* Diagnosa

*Sequence diagram* diagnosa merupakan kegiatan *admin* dalam mendiagnosa penyakit pada balita. Berikut adalah *sequence diagram* dibawah ini.



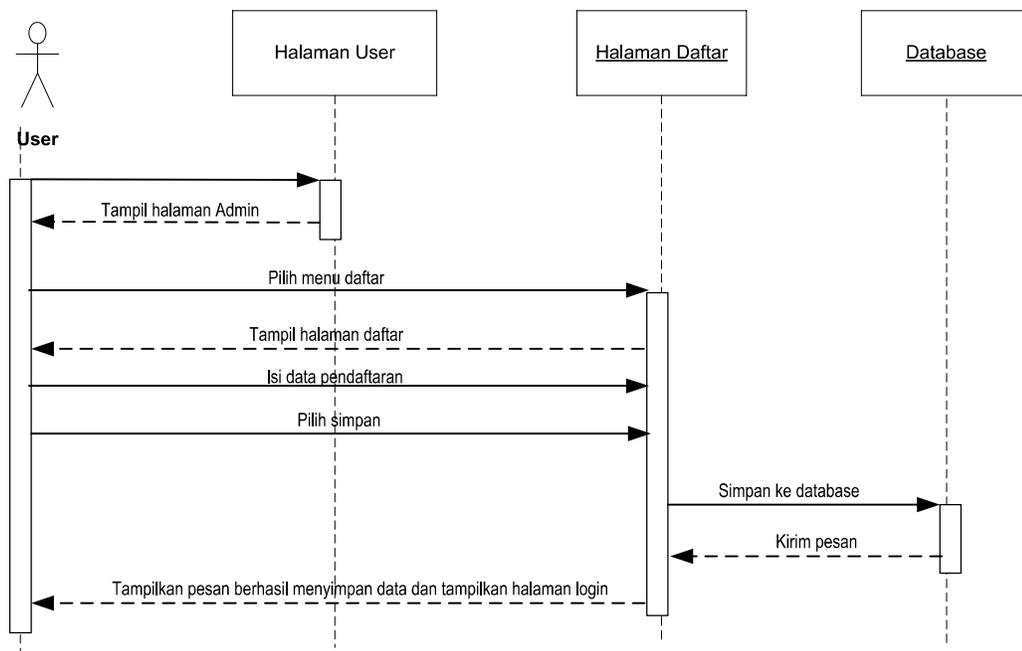
**Gambar 3.21** *Sequence Diagram* Diagnosa  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.21 diatas, *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan pilih menu diagnosa, sistem akan menampilkan halaman diagnosa, kemudian tampil pertanyaan gejala, kemudian pilih jenis gejala penyakit pada anak, selanjutnya cocokkan dengan relasi gejala, lalu akan tersimpan ke *database*

sementara, kemudian tampil halaman hasil diagnosa, setelah itu pilih print hasil diagnosa, maka akan muncul *print* hasil diagnosa.

i. *Sequence Diagram* Daftar

*Sequence diagram* daftar merupakan kegiatan *admin* dalam mengelola dan memilih daftar pada *web* sistem pakar. Berikut adalah *sequence diagram* dibawah ini.



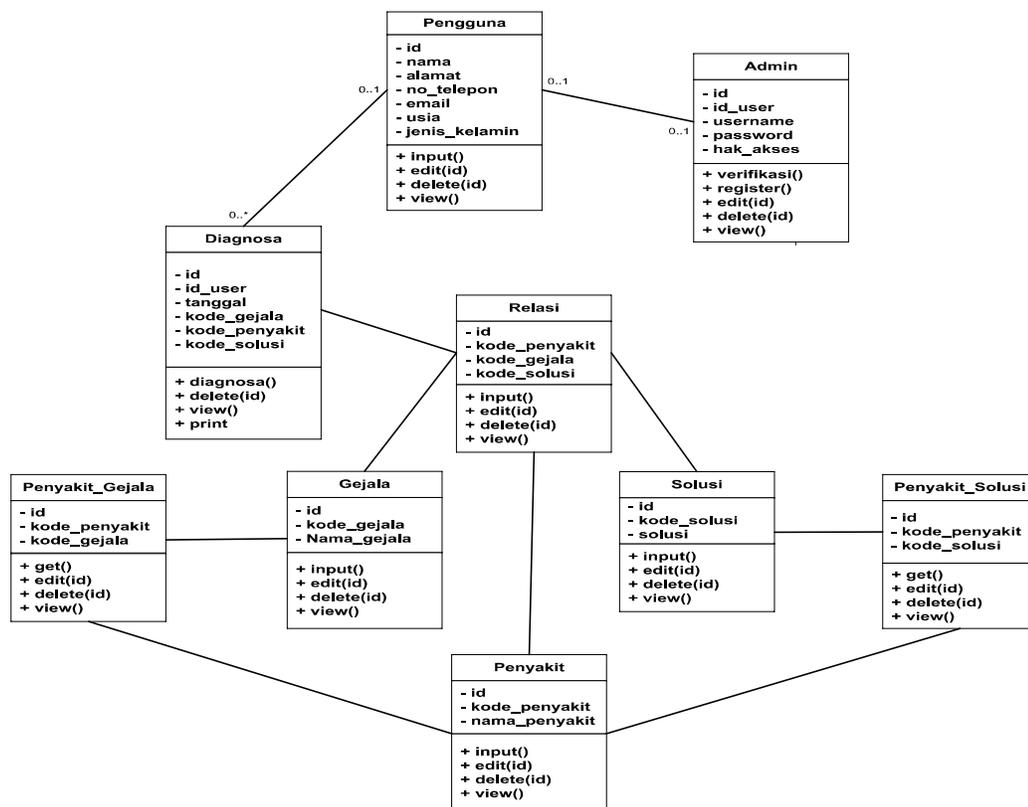
**Gambar 3.22** *Sequence Diagram* Diagnosa  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.22 diatas, *admin* mengakses *web* sistem pakar, kemudian *web* akan menampilkan halaman utama *admin*. Kemudian *admin* menampilkan

pilih menu daftar, sistem akan menampilkan halaman daftar, kemudian tampil isi data pendaftaran dan pilih simpan, kemudian data tersimpan di database dan sistem akan menampilkan halaman *login*.

#### 4. Class Diagram Sistem Pakar

Dalam penelitian sistem pakar penyakit pada balita menjelaskan gambar class diagram hubungan antar class dalam sistem yang telah di bangun dan bagaimana class tersebut berkolaborasi untuk mencapai suatu tujuan. Berikut adalah gambar class diagram yang digunakan pada penelitian ini.

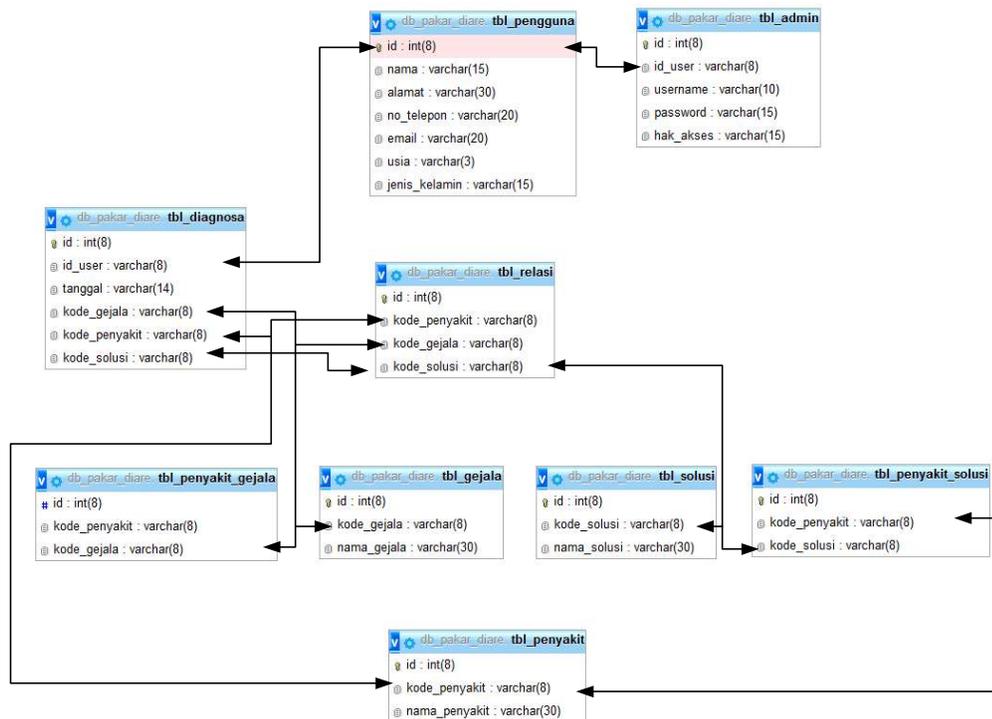


**Gambar 3.23** *Class Diagram*  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.23 diatas, terdapat sembilan *class* yang saling berhubungan, struktur *class diagram* yang pertama adalah *admin* memiliki atribut yaitu *id*, *id user*, *username*, *password* dan hak akses serta memiliki verifikasi input, *edit*, *delete* dan *view*, kemudian pengguna memiliki atribut yaitu *id*, nama, alamat, no telepon, *email*, usia, jenis kelamin dan memiliki fungsi *input*, *edit*, *delete* dan *view*. Kemudian *class diagnosa* memiliki atribut *id*, *id user*, tanggal, kode gejala, kode penyakit, kode solusi dan memiliki fungsi *diagnosa*, *delete*, *view* dan *print*. Kemudian *class relasi* memiliki berupa atribut yaitu *id*, kode penyakit, kode gejala dan kode solusi serta memiliki fungsi *input*, *edit*, *delete*, *view*. Kemudian *class gejala* memiliki berupa atribut yaitu *id*, kode gejala dan nama gejala serta memiliki fungsi *input*, *edit*, *delete*, *view*. Kemudian *class penyakit* memiliki berupa atribut yaitu *id*, kode penyakit, kode gejala dan memiliki fungsi *get*, *edit*, *delete*, *view*. Kemudian *class solusi* memiliki berupa atribut yaitu *id*, kode solusi, solusi dan memiliki fungsi *input*, *edit*, *delete*, *view*. Kemudian *class penyakit solusi* memiliki berupa atribut yaitu *id*, kode penyakit, kode solusi dan memiliki fungsi *input*, *edit*, *delete*, *view*. Kemudian *class penyakit* memiliki berupa atribut yaitu *id*, kode penyakit, nama penyakit dan memiliki fungsi *input*, *edit*, *delete*, *view*.

## 5. Desain Database

Menurut PDM (Physical Data Model) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antar data. Berikut adalah konsep database yang dibuat pada penelitian ini:



**Gambar 3.24** Desain Database  
**Sumber:** Data Penelitian (2018)

Pada gambar 3.24 diatas, terdapat beberapa tabel yang terdiri dari tabel *admin* menyimpan kode *user*, tabel *user* menyimpan data-data *user*, tabel *diagnosa* menyimpan hasil diagnosa penyakit, gejala dan solusi, tabel *relasi* menyimpan kode penyakit, kode gejala dan kode solusi, tabel *penakit gejala* menyimpan kode penyakit dan gejala solusi, tabel *gejala* menyimpan kode gejala dan nama gejala, tabel *penakit solusi* menyimpan kode penyakit dan kode solusi, tabel *solusi* menyimpan kode solusi nama solusi, tabel *penakit* yaitu menyimpan kode penyakit dan nama penyakit.

### 3.4.3 Desain Antar Muka

Desain antar muka yang akan dibuat pada aplikasi sistem pakar mendiagnosis penyakit pada balita:

1. Rancangan *Form Login User*

**Gambar 3.25** *Form Login User*  
**Sumber:** Data Penelitian (2018)

Gambar 3.25 di atas, *form login user* digunakan untuk *Login admin* dan *User* dengan cara *input username* dan *password* kemudian *Login* kehalaman *web* sistem pakar.

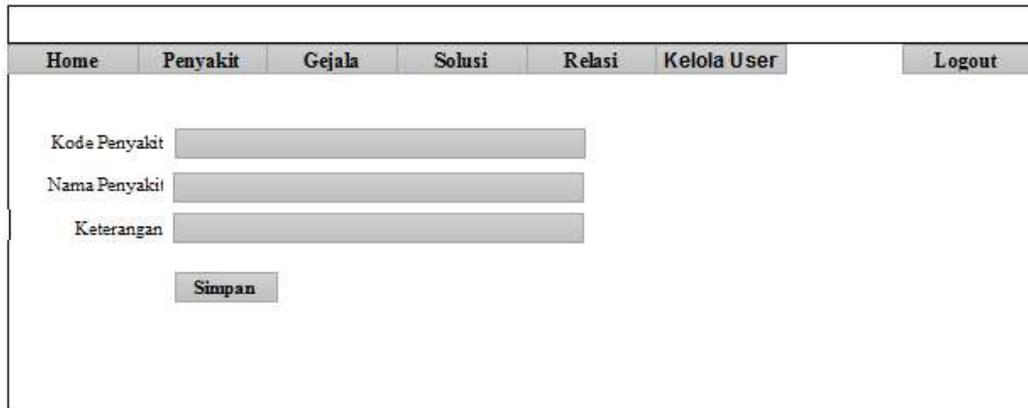
2. Rancangan *Form Penyakit*

Home   Penyakit   Gejala   Solusi   Relasi   Kelola User   Logout				
Add				
+				
NO	Kode Penyakit	Nama Penyakit	Keterangan	Action
1	P01	DENQUE HAERMORHAGIC FEVER		Edit   Delete
2	P02	KEJANG DEMAM		Edit   Delete
3	P03	ISPA		Edit   Delete
4	P04	DEMAM THYPOID		Edit   Delete
5	P05	DIARE		Edit   Delete

**Gambar 3.26** *Form Penyakit*  
**Sumber:** Data Penelitian (2018)

Gambar 3.26 di atas, *form Penyakit* adalah Halaman tampil master penyakit. Halaman ini berfungsi untuk menampilkan semua data master penyakit. Halaman ini hanya bisa dilihat oleh *admin*.

### 3. Rancangan *Form* Tambah Penyakit



Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
------	----------	--------	--------	--------	-------------	--------

Kode Penyakit

Nama Penyakit

Keterangan

**Gambar 3.27** *Form* Tambah Penyakit

**Sumber:** Data Penelitian (2018)

Gambar 3.27 di atas, *form* Tambah Penyakit adalah Halaman tambah data master penyakit. Halaman ini berfungsi untuk menambah data master penyakit.

Halaman ini hanya bisa digunakan oleh *admin*.

### 4. Rancangan *Form* Edit Penyakit

**Gambar 3.28** *Form* Edit Penyakit  
**Sumber:** Data Penelitian (2018)

Gambar 3.28 di atas, *form* Edit Penyakit adalah halaman edit master penyakit. Halaman ini berfungsi untuk mengedit data master penyakit. Halaman ini hanya bisa digunakan oleh *admin*.

#### 5. Rancangan *Form* Gejala

NO	Kode Gejala	Nama Gejala	Keterangan	Action
1	G01	SUHU TINGGI		Edit Delete
2	G02	NYERI DI BELAKANG KEPALA		Edit Delete
3	G03	SUARA SERAK		Edit Delete
4	G04	BATUK		Edit Delete
5	G05	EPITAKSIS		Edit Delete

**Gambar 3.29** *Form* Edit Gejala  
**Sumber:** Data Penelitian (2018)

Gambar 3.29 di atas, *form* Gejala adalah Halaman tampil master gejala. Halaman ini berfungsi untuk menampilkan semua data master gejala. Halaman ini hanya bisa digunakan oleh *admin*.

#### 6. Rancangan *Form* Tambah Gejala

Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
Kode Gejala <input type="text"/> Nama Gejala <input type="text"/> Keterangan <input type="text"/> <input type="button" value="Simpan"/>						

**Gambar 3.30** *Form Tambah Gejala*

**Sumber:** Data Penelitian (2018)

Gambar 3.30 di atas, *form* Tambah Gejala adalah Halaman tambah master gejala. Halaman ini bertujuan untuk menambah data master gejala. Halaman ini hanya bisa digunakan oleh *admin*.

#### 7. Rancangan *Form* Edit Gejala

Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
Kode Gejala <input type="text"/> Nama Gejala <input type="text"/> Keterangan <input type="text"/> <input type="button" value="Simpan"/>						

**Gambar 3.31** *Form Edit Gejala*

**Sumber:** Data Penelitian (2018)

Gambar 3.31 di atas, *form* Edit Gejala adalah halaman *edit* master gejala. Halaman ini berfungsi untuk mengedit data master gejala. Halaman ini hanya bisa digunakan oleh *admin*.

#### 8. Rancangan *Form* Solusi

Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
Add						
NO	Kode Solusi	Nama Solusi	Keterangan	Action		
1	S01	TIRAH BARING		Edit	Delete	
2	S02	DIET MAKAN LUNAK		Edit	Delete	
3	S03	MINUM BANYAK		Edit	Delete	
4	S04	PEMBERIAN ANTI BIOTIK		Edit	Delete	
5	S05	IMMUNISASI		Edit	Delete	

**Gambar 3.32** *Form Solusi*  
**Sumber:** Data Penelitian (2018)

Gambar 3.32 di atas, *formSolusi* adalah Halaman tampil data master solusi. Halaman ini berfungsi untuk menampilkan data master solusi. Halaman ini hanya bisa digunakan oleh *admin*.

#### 9. Rancangan *Form* Tambah Solusi

Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
Kode Solusi <input type="text"/>						
Nama Solusi <input type="text"/>						
Keterangan <input type="text"/>						
Simpan						

**Gambar 3.33** *Form Tambah Solusi*  
**Sumber:** Data Penelitian (2018)

Gambar 3.33 di atas, *form Tambah Solusi* adalah Halaman tambah data master solusi. Halaman ini berfungsi untuk menampilkan semua data master solusi. Halaman ini hanya bisa digunakan oleh *admin*.

#### 10. Rancangan *Form* Edit Solusi

**Gambar 3.34** *Form* Edit Solusi  
**Sumber:** Data Penelitian (2018)

Gambar 3.34 di atas, *form* Edit Solusi adalah Halaman edit master solusi. Halaman ini bertujuan untuk melakukan proses edit data master solusi. Halaman ini hanya bisa digunakan oleh admin.

#### 11. Rancangan *Form* Relasi

NO	Penyakit	Gejala	Solusi	Action
1				Edit Delete
2				Edit Delete
3				Edit Delete
4				Edit Delete
5				Edit Delete

**Gambar 3.35** *Form* Relasi  
**Sumber:** Data Penelitian (2018)

Gambar 3.35 di atas, *form* Relasi adalah Halaman tampil relasi. Halaman ini berfungsi untuk menampilkan data relasi antara penyakit dengan gejala dan penyakit dengan solusi. Halaman ini hanya bisa dilihat oleh admin.

#### 12. Rancangan *Form* Relasi Gejala

**Gambar 3.36** *Form Relasi Gejala*

**Sumber:** Data Penelitian (2018)

Gambar 3.36 di atas, *form* relasi gejala adalah halaman relasi gejala. Halaman ini berfungsi untuk membuat relasi antar penyakit dengan gejala. Halaman ini hanya bisa digunakan oleh *admin*.

### 13. Rancangan *Form Relasi Solusi*

**Gambar 3.37** *Form Relasi Solusi*

**Sumber:** Data Penelitian (2018)

Gambar 3.37 di atas, *form* Relasi Solusi adalah Halaman relasi solusi. Halaman ini berfungsi untuk membuat relasi antara penyakit dan solusi. Halaman ini hanya bisa digunakan oleh *admin*.

### 14. Rancangan *Form Kelola User*

Home	Penyakit	Gejala	Solusi	Relasi	Kelola User	Logout
<input type="button" value="Add"/>						
NO	Kode User	Username	Nama	Action		
1				<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	
2				<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	
3				<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	
4				<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	
5				<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	

**Gambar 3.38** *Form* Kelola User  
**Sumber:** Data Penelitian (2018)

Gambar 3.38 di atas, *form* Relasi Solusi adalah Halaman kelola user. Halaman ini berfungsi untuk menampilkan data semua user yang ada. Halaman ini hanya bisa dilihat oleh admin.

#### 15. Rancangan *Form* Tambah User

Home Penyakit Gejala Solusi Relasi Kelola User Logout

Username

Nama

Password

Hak Akses

Simpan

**Gambar 3.39** *Form Tambah User*

**Sumber:** Data Penelitian (2018)

Gambar 3.39 di atas, *form Tambah User* adalah Halaman tambah user. Halaman ini berfungsi untuk menambah user. Halaman ini hanya bisa digunakan oleh admin.

#### 16. Rancangan *Form Diagnosa*

Home Penyakit Gejala Solusi Relasi Kelola User Logout

Pilih Gejala pada Anak Balita

Simpan

**Gambar 3.40** *Form Diagnosa*

**Sumber:** Data Penelitian (2018)

Gambar 3.40 di atas, *form Diagnosa* adalah Halaman diagnosa. Halaman ini berfungsi untuk melakukan diagnosa penyakit anak. Dengan memilih data-data

gejala yang terjadi pada anak. Kemudian klik lanjutkan untuk melihat jenis penyakit dan solusinya. Halaman ini digunakan oleh user.

### 17. Rancangan *Form* Profil

**Gambar 3.41** *Form* Profil  
**Sumber:** Data Penelitian (2018)

Gambar 3.41 di atas, *form* Profil adalah Halaman yang berfungsi untuk mengganti *password* user atau admin.

### 18. Rancangan *Form* Data Diagnosa

NO	Tanggal	Penyakit	Solusi	Action
1	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	Detail Delete
2	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	Detail Delete
3	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	Detail Delete
4	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	Detail Delete
5	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	ZZZ ZZZZZ ZZZ	Detail Delete

**Gambar 3.42** *Form* Data Diagnosa  
**Sumber:** Data Penelitian (2018)

Gambar 3.42 di atas, *form* Data Diagnosa adalah Halaman yang berfungsi untuk menampilkan data hasil diagnosa user. Data yang ditampilkan berdasarkan data user yang login.

#### 19. Rancangan *Form* Halaman *Home*



**Gambar 3.43** *Form* Halaman *Home*  
**Sumber:** Data Penelitian (2018)

Gambar 3.43 di atas, *form* Halaman *Home* adalah halaman yang menampilkan informasi tentang aplikasi sistem pakar mendiagnosa penyakit pada Balita.

### 3.5 Lokasi & Jadwal Penelitian

#### 3.5.1 Lokasi Penelitian

Dalam melakukan penelitian ini lokasi tempat penelitian di Rs.Awal Bros. Sebagai narasumber pakar untuk penelitian ini adalah Hartuti S.ST dilakukan berdasarkan data-data yang didapatkan dari pihak terkait dengan penelitian ini.

### 3.5.2 Jadwal Penelitian

Jadwal penelitian untuk memperoleh data dan informasi dilaksanakan pada bulan november 2017 Sedangkan waktu penelitian ini disesuaikan dengan waktu senggang pembelajaran atau jam tertentu. Berikut jadwal penelitian selengkapnya.

**Tabel 3.7** Jadwal Penelitian

No.	Kegiatan	Bulan 2017/2018																			
		Maret-Mei 2018				Juni- September 2018				November- Desember 2018				Januari 2019				Februari 2019			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul Penelitian	■	■																		
2	Penulisan BAB I			■	■																
3	Penulisan BAB II			■	■	■	■	■													
5	Penulisan BAB III							■	■	■	■										
6	Penulisan BAB IV											■	■	■	■						
7	Hasil BAB V Daftar isi, Lampiran															■	■	■	■		

Sumber : Pengolahan Data Penelitian (2018)