

**SISTEM PAKAR DIAGNOSIS PENYAKIT PENCERNAAN
MENGUNAKAN METODE FORWARD
CHAINING BERBASIS WEB**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh
Kelulusan jenjang Strata Satu (SI) pada
Program Studi Teknik Informatika

Disusun Oleh :

Adnan Hafizh
361601005



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI
2021**

LEMBAR PENGESAHAN

SISTEM PAKAR DIAGNOSIS PENYAKIT PENCERNAAN MENGUNAKAN METODE FORWARD CHAINING BERBASIS WEB

Oleh :

Adnan Hafizh
361601005

Tugas Akhir Ini Telah Diterima Dan Disahkan Untuk Memenuhi Persyaratan
Mencapai Gelar Sarjana Teknik Informatika

Pada

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

Bandung, 14 Februari 2021
Disetujui Oleh

Ketua Program Studi

Dosen Pembimbing

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

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

LEMBAR PERSETUJUAN REVISI

SISTEM PAKAR DIAGNOSIS PENYAKIT PENCERNAAN MENGUNAKAN METODE FORWARD CHAINING BERBASIS WEB

Oleh :

Adnan Hafizh
361601005

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

Bandung, 14 Januari 2021

Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	Chalifa Chazar, S.T., M.T.	Dosen Pembimbing	
2	Hendra Gunawan, S.T., M.Kom.	Penguji 1	
3	Dede Suryana, Ir., M.Si.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika

Chalifa Chazar.,S.T., M.T.
NIDN. 0412098701

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa :

- 1) Tugas akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik. Baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- 2) Tugas akhir ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terimakasih.
- 3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya seni saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

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

Bandung, 28 Februari 2021

Yang Membuat Pernyataan

Adnan hafizh
361601005

ABSTRAK

SISTEM PAKAR DIAGNOSIS PENYAKIT PENCERNAAN MENGUNAKAN METODE FORWARD CHAINING BERBASIS WEB

Oleh :

Adnan Hafizh
361601005

Dikarenakan kesadaran akan kesehatan masyarakat yang masih rendah, kebiasaan hidup dari masyarakat yang selalu ingin hidup praktis, pengetahuan masyarakat yang sedikit akan gejala-gejala awal dari suatu penyakit pencernaan merupakan salah satu faktor penyebab penyakit pencernaan menjadi parah ketika pasien ditangani oleh tenaga paramedis. Untuk mengantisipasi hal tersebut peneliti berusaha membangun sistem pakar untuk mendiagnosis penyakit pencernaan berdasarkan gejala-gejala yang diderita oleh pasien. Sistem pakar yang dibangun oleh peneliti menggunakan metode *Forward Chaining* sebagai metode palacakannya. Dibangunnya sistem pakar diagnosis penyakit pencernaan diharapkan dapat memberikan informasi kepada pasien dengan cara yang mudah dan tidak terbatas oleh waktu.

Kata Kunci : Penyakit, Pencernaan, Diagnosa

ABSTRACT

WEB BASED DIGESTIVE DISEASE DIAGNOSYS USING FORWARD CHAINING EXPERT SYSTEM

Oleh :

**Adnan Hafizh
361601005**

Due to the low awareness of public health, the habits of people who always want to live practically, the community's little knowledge of the early symptoms of a digestive disease is one of the factors that cause digestive diseases to become severe when patients are treated by paramedics. To anticipate this, researchers are trying to build an expert system to diagnose digestive diseases based on the symptoms suffered by the patient. The expert system built by the researcher uses the Forward Chaining method as a tracking method. The development of an expert system for diagnosing digestive diseases is expected to provide information to patients in an easy and time-limited way.

Keywords : Disease, Digestion, Diagnose

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT karena atas berkat limpahan Rahmat dan Karunia-Nya, penulis bisa menyelesaikan laporan penelitian tugas akhir yang berjudul, **“SISTEM PAKAR DIAGNOSIS PENYAKIT PENCERNAAN MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS *WEB*”**, sesuai dengan yang direncanakan.

Tugas akhir ini berisi mengenai perancangan sebuah aplikasi sistem pakar yang bertujuan untuk memberi gambaran awal penyakit berdasarkan gejala yang dialami, yang meliputi analisis dari sistem yang diusulkan dengan harapan dapat mengatasi masalah yang diidentifikasi.

Dengan segala keterbatasan tentunya diharapkan aplikasi ini dapat bermanfaat bagi berbagai pihak, khususnya bagi penulis sendiri.

Bandung, 14 Februari 2021

Adnan Hafizh
361601005

UCAPAN TERIMAKASIH

Dengan mengucapkan syukur Alhamdulillah, penelitian ini dapat diselesaikan untuk memenuhi syarat tugas akhir. Laporan penelitian dalam tugas akhir ini diajukan untuk memenuhi dan melengkapi salah satu syarat akademik dalam kelulusan jenjang Strata Satu (S1) jurusan Teknik Informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Penyusunan tugas akhir ini tidak lepas dari dukungan dan bimbingan dari berbagai pihak, maka pada kesempatan ini penulis ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada :

1. Ibu Chalifa Chazar, S.T., M.T. selaku Dosen pembimbing yang selalu meluangkan waktu, pikiran dan tenaga dalam memberikan bimbingan, masukan dan saran-sarannya dan selaku Ketua program studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
2. Bapak Dr. Chairuddin, M.T., M.M. selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM).
3. Seluruh Dosen, Staff dan Karyawan Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri (STMIK-IM) yang telah mendidik dan membantu dalam memberika informasi serat motivasi dalam proses studi maupun tugas akhir berlangsung.
4. Teruntuk Kedua Orang Tua Tercinta Bapak Epmal Permadi dan Ibu Hany Yuningsih yang sangat penulis sayangi dan cintai. Terimakasih selalu

memberikan nasehat, dukungan, didikan, kasih sayang, serta Do'a yang penuh dan tulus.

5. Teruntuk kakak penulis Rakha Ghatfan Muhadzib. Terima kasih selalu memberikan dukungan, kasih sayang, serta Do'a yang tulus.
6. kakek, nenek, paman, bibi, dan serta keluarga besar penulis, yang sangat penulis sayangi dan cintai, terimakasih selalu telah mendukung, dan mendo'akan penulis selalu.
7. Sahabat-sahabat penulis yaitu Aswin Winangga, Alvin Surya Ramadhan, Irsan Hamdani, Rezki Dwi yang sama-sama berjuang untuk menyelesaikan laporan penelitian tugas akhir ini.
8. Seluruh rekan STMIK-IM angkatan 2016 khususnya pada program studi Teknik Informatika yang sama-sama berjuang untuk terus meraih impian, yang saling membantu juga memberikan semangat, dan seringkali menjadi tempat sharing, baik yang berhubungan dengan materi perkuliahan maupun lainnya.
9. Seluruh rekan sahabat, dan pihak-pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu Penulis baik secara langsung maupun tidak langsung memberikan semangat kepada Penulis dalam menyelesaikan laporan penelitian tugas akhir.

Penulis menyadari bahwa masih banyak kekurangan yang mendasar pada laporan penelitian tugas akhir ini. Oleh karena itu penulis mengundang pembaca untuk memberikan saran serta kritik yang dapat membangun penulis. Penulis berharap adanya kritik konstruktif dan saran yang membangun dari semua pihak.

Akhir kata saya, berharap semoga dengan selesainya laporan penelitian Tugas Akhir ini dapat memberikan manfaat bagi semua pihak serta menambah wawasan bagi pemikiran kita semua. Terimakasih.

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN REVISI.....	ii
SURAT PERNYATAAN.....	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	4
1.5 Metode Penelitian.....	4
1.5.1 Teknik Pengumpulan Data.....	4
1.5.2 Metode Perancangan.....	5
1.6 Sistematika Penulisan.....	7

BAB II LANDASAR TEORI	8
2.1 Sistem	8
2.1.1 Elemen Sistem	8
2.1.2 Karakteristik Sistem.....	8
2.2 Sistem Pakar	10
2.2.1 Keuntungan dan Kelemahan Sistem Pakar	12
2.2.2 Tujuan Sistem Pakar	13
2.2.3 Struktur Sistem Pakar	13
2.3 Diagnosis	16
2.4 Penyakit	17
2.5 Pencernaan.....	17
2.6 Penyakit Pencernaan.....	17
2.7 Pengertian Web	17
2.8 Pengertian Berbasis Web.....	18
2.9 Metode Forward Chaining.....	18
2.10 Basis Data.....	19
2.10.1 <i>My Structured Query Language (MySQL)</i>	19
2.10.2 <i>Structured Query Language (SQL)</i>	19
2.11 Bahasa Pemrograman	20
2.11.1 <i>Hypertext Preprocessor (PHP)</i>	20
2.11.2 <i>Hypertext Markup language (HTML)</i>	20
2.11.3 <i>Java Script</i>	21

2.11.4	<i>Cascading Style Sheet (CSS)</i>	21
2.12	Aplikasi Pendukung	21
2.12.1	<i>PhpMyAdmin</i>	21
2.12.2	<i>Xampp</i>	22
2.13	<i>Enterprise Relationship Diagram (ERD)</i>	23
2.14	<i>Flowchart</i>	28
2.14.1	Jenis <i>Flowchart</i>	28
2.14.2	Simbol <i>Flowchart</i>	29
2.15	<i>Data Flow Diagram (DFD)</i>	30
2.16	Metode Pengembangan	34
BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM.....		37
3.1	<i>Communication</i>	37
3.1.1	Survei.....	37
3.1.2	Studi Literatur	37
3.2	<i>Planning</i>	40
3.2.1	Penjadwalan	40
3.3	<i>Modelling</i>	42
3.3.1	Analisis Sistem	42
3.3.2	Perancangan Sistem	51
3.3.3	<i>Design Interface</i>	62
BAB IV IMPLEMENTASI DAN UJI COBA		74
4.1.	<i>Construction</i>	74

4.1.1 Implementasi <i>Hardware & Software</i>	74
4.1.2 Implementasi Antarmuka.....	75
4.1.3 <i>Testing</i>	80
4.1.4 Kesimpulan <i>Testing</i>	92
BAB V PENUTUP.....	95
5.1 Kesimpulan.....	95
5.2 Saran.....	95
DAFTAR PUSTAKA	97
LAMPIRAN.....	101

DAFTAR TABEL

Tabel 2.3 Simbol-Simbol <i>Flowchart</i>	29
Tabel 2.4 Simbol-simbol <i>Data Flow Diagram</i>	31
Tabel 3.1 Referensi Penelitian	38
Tabel 3.2. Penjadwalan Penelitian	41
Tabel 3.3. Tabel Penyakit Pencernaan	46
Tabel 3.4. Tabel Gejala Penyakit Pencernaan.....	47
Tabel 3.5. Tabel Basis Pengetahuan	48
Tabel 3.6. Tabel Admin	60
Tabel 3.7. Tabel Penyakit.....	61
Tabel 3.8. Tabel Gejala	61
Tabel 3.9. Tabel Basis Pengetahuan	61
Tabel 4.1. Tabel Rencana Pengujian.....	81
Tabel 4.2. Tabel Hasil Pengujian <i>Login Admin</i>	82
Tabel 4.3. Tabel Hasil Pengujian Tambah Data Penyakit.....	83
Tabel 4.4. Tabel Hasil Pengujian Ubah Data Penyakit	84
Tabel 4.5. Tabel Hasil Pengujian Hapus Data Penyakit	84
Tabel 4.6. Tabel Hasil Pengujian Tambah Data Gejala	85

Tabel 4.7. Tabel Hasil Pengujian Ubah Data Gejala	86
Tabel 4.8. Tabel Hasil Pengujian Hapus Data Gejala.....	87
Tabel 4.9. Tabel Hasil Pengujian Tambah Data Basis Pengetahuan	87
Tabel 4.10. Tabel Hasil Pengujian Ubah Data Basis Pengetahuan.....	88
Tabel 4.11. Tabel Hasil Pengujian Hapus Data Basis Pengetahuan	89
Tabel 4.12. Tabel Hasil Pengujian Diagnosa	90
Tabel 4.13. Tabel rencana <i>Testing</i> Klasifikasi Penyakit Pencernaan.....	92

DAFTAR GAMBAR

Gambar 1.1. Waterfall Pressman (Pressman, 2015).....	5
Gambar 2.1. Struktur Sistem Pakar	14
Gambar 2.2. Kardinalitas Relasi Satu ke Satu (<i>One to One</i>)	25
Gambar 2.3. Kardinalitas Relasi Satu ke Banyak (<i>One to Many</i>).....	26
Gambar 2.4. Kardinalitas Relasi Banyak ke Satu (<i>Many to One</i>).....	26
Gambar 2.5. Kardinalitas Relasi Banyak ke Banyak (<i>Many to Many</i>)	27
Gambar 2.6. Waterfall Pressman (Pressman, 2015).....	34
Gambar 3.1. Sistem yang Sedang Berjalan	43
Gambar 3.2. Sistem yang Diusulkan.....	45
Gambar 3.3. Pohon Pelacakan	50
Gambar 3.3. Context Diagram	52
Gambar 3.4. DFD Level 1 Sistem Pakar Diagnosa Penyakit Pencernaan	53
Gambar 3.5. DFD Level 2 Proses 1.0 (Proses Login).....	54
Gambar 3.7. DFD Level 2 Proses 3.0 (Proses Diagnosa)	56
Gambar 3.8. DFD Level 3 Proses 2.1 (Mengelola Data Penyakit)	57
Gambar 3.9. DFD Level 3 Proses 2.2 (Mengelola Data Gejala).....	58
Gambar 3.10. DFD Level 3 Proses 2.3 (Mengelola Data Basis Pengetahuan)	59

Gambar 3.11. ERD Sistem Pakar Diagnosa Penyakit Pencernaan	60
Gambar 3.12. <i>Design Interface</i> Halaman Utama	62
Gambar 3.13. <i>Design Interface</i> Halaman About.....	63
Gambar 3.14. <i>Design Interface</i> Halaman Diagnosa.....	64
Gambar 3.15. Design Interface Halaman Login	65
Gambar 3.16. <i>Design Interface</i> Halaman Utama Admin.....	66
Gambar 3.17. <i>Design Interface</i> Halaman <i>Rule</i> Admin	67
Gambar 3.18. <i>Design Interface</i> Halaman Gejala Admin	68
Gambar 3.19. <i>Design Interface</i> Halaman Penyakit Admin.....	69
Gambar 3.20. Design Interface Halaman Update Rule.....	70
Gambar 3.21. <i>Design Interface</i> Halaman <i>Update</i> Gejala	71
Gambar 3.22. <i>Design Interface</i> Halaman <i>Update</i> Penyakit.....	72
Gambar 3.23. Design Interface Halaman Tambah Penyakit.....	73
Gambar 4.1. Tampilan Menu Utama.....	75
Gambar 4.2. Tampilan Menu About	76
Gambar 4.3. Tampilan Menu Diagnosa	76
Gambar 4.4. Tampilan Menu Hasil Diagnosa.....	77
Gambar 4.5. Tampilan Menu Login Admin.....	78
Gambar 4.6. Tampilan Menu Admin	78
Gambar 4.7. Tampilan Menu <i>Rule</i>	79

Gambar 4.8. Tampilan Menu Gejala.....	79
Gambar 4.9. Tampilan Menu Penyakit	80

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kesehatan adalah hal terpenting dalam kehidupan, menjaga pola hidup tetap sehat dapat membuat tubuh terhindar dari penyakit dan membuat sistem yang berada di tubuh kita dapat bekerja dengan optimal, salah satu sistemnya adalah sistem pencernaan. Sistem ini membutuhkan asupan makanan yang bergizi agar dapat merubah makanan tersebut menjadi energi yang dibutuhkan tubuh. Semakin baik makanan dan pola hidup yang kita lakukan, semakin baik pula energi yang akan dihasilkan oleh sistem pencernaan kita, begitu juga sebaliknya, namun terkadang banyak orang masih tidak terlalu memperhatikan kesehatannya sehingga timbul lah penyakit pencernaan (Situmorang & Damantalm, dkk, 2016).

Penyakit pencernaan merupakan penyakit yang menyerang organ pencernaan sehingga mengganggu kerja sistem pencernaan. Beberapa faktor yang dapat menyebabkan penyakit pencernaan antara lain makanan yang kurang baik, keseimbangan nutrisi, pola makan yang tidak teratur, dan infeksi serta kelainan pada organ pencernaan. Orang yang terkena penyakit ini akan mengalami gangguan pencernaan seperti diare, maag, dan lain-lain (Ma'rifati & Kesuma, 2018).

Penyakit pada saluran pencernaan merupakan penyakit yang berbahaya dan banyak menyebabkan kematian. Berdasarkan data dari WHO (*World Health Organization*), penyakit pada saluran pencernaan, diantaranya kanker usus

merupakan penyakit yang paling banyak menyebabkan kematian nomor 6 di dunia, dan penyakit diare merupakan penyakit yang menyebabkan kematian nomor 7 di dunia. Sistem pencernaan pada manusia merupakan salah satu organ vital bagi tubuh, sehingga kesehatan sistem pencernaan sangatlah penting untuk dijaga. Mengingat fungsi dari sistem pencernaan sebagai tempat atau alat untuk mencerna setiap makanan dan minuman yang masuk ke dalam tubuh manusia. Kesadaran akan kesehatan masyarakat yang masih rendah, kebiasaan hidup dari masyarakat yang selalu ingin hidup praktis, perilaku dan pola pikir yang cenderung mengarah bergaya hidup tidak sehat, pengetahuan masyarakat yang sedikit akan gejala-gejala awal dari suatu penyakit merupakan salah faktor-faktor penyebab penyakit menjadi parah ketika penderita ditangani oleh tenaga paramedis, sehingga perlu adanya suatu sistem untuk mendiagnosa penyakit pada saluran pencernaan beserta saran atau solusi yang diperlukan sesuai medis (Istiqomah & Fadlil, 2013).

Maka perlu dibuat aplikasi sistem pakar untuk mendiagnosis penyakit dalam khususnya penyakit pencernaan. Dimana aplikasi ini diberikan pengetahuan-pengetahuan sebelumnya dalam mendiagnosa suatu penyakit, dengan memberikan suatu pernyataan dan informasi kepada sistem, maka sistem akan mengambil kesimpulan dengan cepat dan tepat akan suatu penyakit yang diderita oleh pasien.

Berdasarkan permasalahan tersebut, penelitian ini akan mencoba membangun suatu sistem pakar yang bekerja seperti seorang pakar yang bisa digunakan untuk mendiagnosa penyakit pada pencernaan. Dalam melakukan diagnosa, sistem pakar membutuhkan sebuah metode, salah satu metode yang digunakan dalam penelitian ini adalah *Forward Chaining*.

Forward Chaining adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi atau fakta yang berupa data, kemudian bergerak maju melalui premis-premis dan penggabungan rule untuk menghasilkan suatu kesimpulan (*booton up reasoning*) (Nur & Salisah, dkk, 2015).

Dari latar belakang diatas, penelitian ini akan mencoba untuk merancang suatu program sistem pakar yang mampu memberikan diagnosis yang akurat akan kemungkinan pencernaan menderita suatu penyakit beserta solusi awal penanganannya. Maka judul penelitian yang akan diambil adalah “**Sistem Pakar Diagnosis Penyakit Pencernaan Menggunakan Metode *Forward Chaining* Berbasis Web**”.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang, masalah yang dikemukakan diatas yang menjadi pokok masalah dalam tugas akhir ini adalah:

1. Bagaimana membangun sistem pakar yang dapat mendiagnosis penyakit pencernaan berdasarkan gejala-gejala yang muncul menggunakan metode *Forward Chaining*.
2. Apakah sistem pakar dapat memberikan informasi, diagnosa dengan cara yang mudah dan tidak terpaut oleh waktu.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitan ini adalah:

1. Dapat menghasilkan sebuah sistem pakar yang dapat mendiagnosis penyakit pencernaan berdasarkan gejala-gejala yang muncul menggunakan metode *Forward Chaining*.

2. Diharapkan dapat memberikan informasi, diagnosa dengan cara yang mudah dan tidak terbatas oleh waktu.

1.4 Batasan Masalah

Agar masalah yang diteliti tidak keluar dan menyimpang, maka diperlukannya suatu batasan masalah tersebut adalah:

1. Penyakit yang diidentifikasi pada penelitian ini hanya penyakit pada pencernaan.
2. Metode *Forward Chaining* diterapkan untuk menentukan hasil identifikasi.
3. Perancangan sistem pakar diagnosis penyakit pencernaan dirancang sebagai aplikasi web.
4. Tahapan dalam metode perancangan hanya sampai tahapan *construction*

1.5 Metode Penelitian

1.5.1 Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang dilakukan penulis adalah:

1. Survei

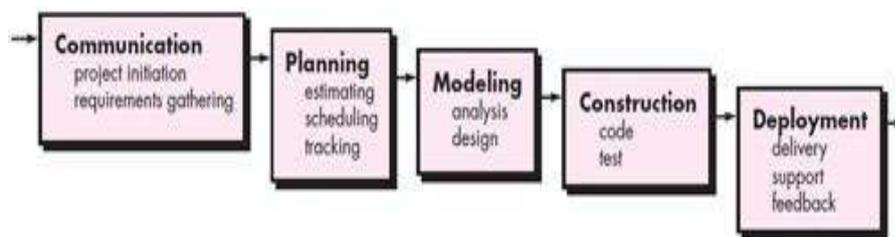
Merupakan penelitian kuantitatif menggunakan pertanyaan-pertanyaan dalam bentuk kuisioner yang akan diberikan kepada 10 responden untuk mengukur validasi data-data yang didapat.

2. Studi Literatur:

Metode dimana informasi yang di peroleh penulis bersumber dari pengumpulan data-data yang di dapat dari berbagai sumber seperti, buku, jurnal, ataupun literature yang berhubungan dengan materi sistem pakar.

1.5.2 Metode Perancangan

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



Gambar 1.1 Waterfall Pressman (Pressman, 2015)

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

1. *Communication*

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

2. *Planning*

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

3. *Modeling*

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

4. *Construction*

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

5. *Deployment*

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

1.6 Sistematika Penulisan

Penulisan skripsi ini terbagi menjadi 5 bab, dengan sistematika sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, identifikasi masalah, tujuan penelitian, batasan masalah, metode penelitian, metode perancangan, dan sistematika penulisan tugas akhir.

BAB II : LANDASAN TEORI

Bab ini memuat tentang konsep dan landasan teori yang berhubungan dengan Sistem Pakar.

BAB III : ANALISA MASALAH DAN PERANCANGAN PROGRAM

Bab ini memuat tentang pembahasan masalah yang akan dipecahkan dengan menggunakan teknik perancangan sistem UML (*Unified Modelling Language*).

BAB IV : IMPLEMENTASI DAN UJI COBA

Bab ini memuat tentang penjelasan dalam pengoperasian program secara bertahap. Serta memuat tentang hasil evaluasi implementasi program, termasuk kelebihan dan kekurangannya. Untuk evaluasi pada bab ini akan di terapkan metode *Forward Chaining*.

BAB V : PENUTUP

Bab ini memuat tentang kesimpulan berdasarkan pembahasan sebelumnya serta saran dari hasil analisis penulis tentang permasalahan yang dibahas.

BAB II

LANDASAR TEORI

2.1 Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul beresama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu .

Penjelasan di atas menjelaskan bahwa sistem bekerja dalam suatu jaringan kerja dari suatu prosedur yang saling berhubungan satu sama lain untuk menyelesaikan tujuan dan sasaran yang dimaksud. Definisi sistem juga dapat dijelaskan oleh Jogiyanto dalam bukunya *Analisa dan Desain Sistem Informasi*, menerangkan: “sistem adalah kumpulan dari elemen elemen yang berinteraksi untuk mencapai suatu tujuan tertentu” (Jogiyanto, 2005).

2.1.1 Elemen Sistem

Elemen sistem adalah suatu sistem terdiri dari sejumlah elemen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan (Jogiyanto, 2005).

2.1.2 Karakteristik Sistem

Sistem memiliki karakteristik atau sifat-sifat yang tertentu yaitu mempunyai komponen (*Components*), batas sistem (*Boundary*), Lingkungan (*Environments*), penghubung (*Interface*), masukan (*input*), keluaran (*Output*), pengolah (*Process*),

dan sasaran (*Objectives*) atau tujuan (*Goals*). Sistem mempunyai karakteristik atau sifat-sifat tertentu (Jogiyanto, 2005), yaitu:

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan.

b. Sifat-Sifat

Sistem itu terdiri untuk menjalankan fungsi tertentu dan mempunyai sistem yang lain secara keseluruhan.

c. Batasan (*Boundary*) Sistem

Merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya

d. Lingkungan Luar Sistem (*Environment*)

Adalah apapun diluar batasan sistem yang mempengaruhi operasi sistem, lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

e. Penghubung (*Interface*) Sistem

Merupakan media penghubung antara suatu subsistem dengan subsistem lainnya.

f. Masukkan Sistem (*Input*)

Adalah energi yang dimasukkan kedalam sistem, masukan dapat berupa masukan perawatan (*Maintenance Input*), dan masukan sinyal (*Signal Input*), *Maintenance Input* adalah energi yang dimasukkan supaya sistem tersebut dapat

beroperasi. Sedangkan *Signal Input* adalah energi yang di proses untuk mendapatkan kelauran.

g. Keluaran Sistem (*Output*)

Adalah hasil energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat merupakan masukan untuk subsistem yang lain kepada supra sistem.

h. Sasaran Sistem

Suatu sistem pasti memiliki sasaran atau tujuan (*Goal*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2 Sistem Pakar

Sistem Pakar (*expert system*) adalah salah satu teknik kecerdasan buatan yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli (Kusumadewi, 2003).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktifitas pemecahan masalah. Beberapa aktifitas pemecahan masalah yang dimaksud seperti (Lestari, 2012):

a. Interpretasi

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi, termasuk pengenalan ucapan, analisis citra, interpretasi sinyal, dll.

b. Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.

Contoh: prediksi demografi, prediksi ekonomi, dll.

c. Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.

d. Perancangan (desain)

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu. Contoh: perancangan layout sirkuit, bangunan.

e. Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.

f. *Monitoring*

Membandingkan hasil pengamatan dengan kondisi yang diharapkan. Contoh: *computer aided monitoring system*.

g. *Debugging*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.

Contoh: memberikan resep obat terhadap kegagalan.

h. Intruksi

Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.

Contoh: melakukan instruksi untuk diagnosis dan *debugging*.

i. Kontrol

Mengatur tingkah laku suatu environment yang kompleks. Contoh: melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

2.2.1 Keuntungan dan Kelemahan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar ahli (Kusumadewi, 2003), antara lain:

- a. Memungkinkan orang awam dapat mengerjakan pekerjaan para ahli.
- b. Dapat melakukan proses secara berulang dan otomatis.
- c. Menyimpan pengetahuan dan keahlian dari para pakar.
- d. Meningkatkan *output* dan produktivitas serta kualitas.
- e. Mampu mengambil dan melestarikan keahlian para pakar (terutama termasuk keahlian yang langka).
- f. Mampu beroperasi dalam lingkungan yang berbahaya.
- g. Memiliki kemampuan untuk mengakses pengetahuan dan bekerja dengan informasi yang kurang lengkap dan kurang pasti atau bahkan tidak lengkap dan tidak pasti.

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan (Kusumadewi, 2003), antara lain:

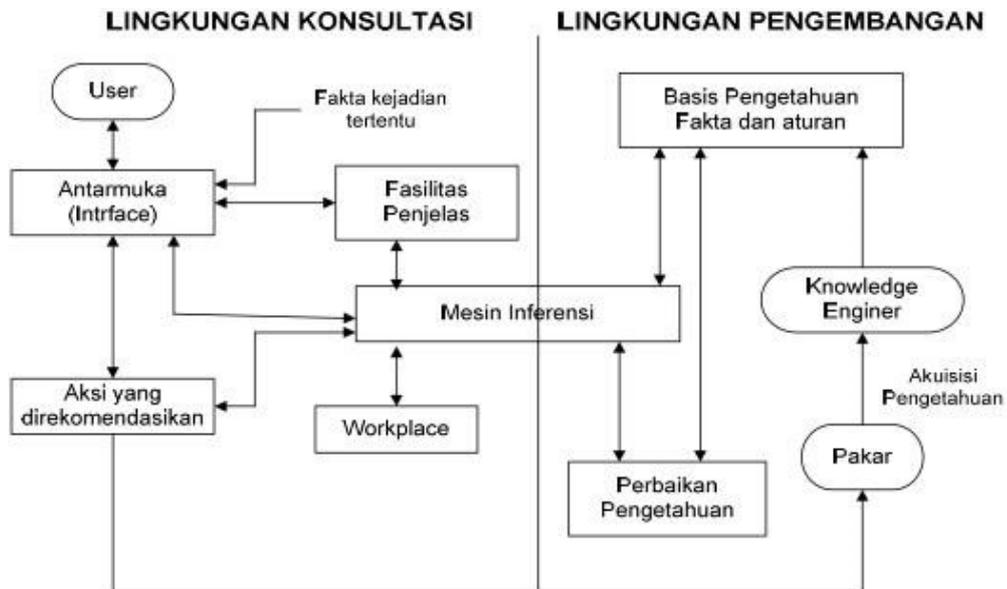
- a. Biaya yang diperlukan untuk pembuatan dan pemeliharaan aplikasi sistem pakar sangat mahal.
- b. Sulit dikembangkan, yang dikarenakan ketersediaan pakar di bidangnya yang terbatas.
- c. Sistem pakar tidak 100% benar, karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu dibutuhkan uji ulang secara teliti sebelum digunakan.

2.2.2 Tujuan Sistem Pakar

Tujuan utama sistem pakar bukan untuk mengganti kedudukan seorang ahli maupun pakar, tetapi untuk memasyarakatkan pengetahuan dan pengalaman pakar-pakar yang ahli dibidangnya (Febriansyah, 2008).

2.2.3 Struktur Sistem Pakar

Sistem pakar terdiri dari dua bagian pokok, yaitu: lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi (Kusumadewi, 2003).



Gambar 2.1 Struktur Sistem Pakar

a. Antarmuka Pengguna

Merupakan mekanisme yang digunakan oleh pengguna dari sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

b. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas 2 (dua) elemen dasar yaitu:

1. Fakta:

Informasi tentang objek dalam area permasalahan.

2. Aturan:

Informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

c. Fasilitas Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya di transfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.

d. Mesin Inferensi

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan satu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.

e. *Workplace*

Workplace merupakan area dari kesimpulan memori kerja (*working memory*) digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 (tiga) keputusan yang dapat direkam:

1. Rencana: bagaimana menghadapi masalah

2. Agenda: aksi-aksi yang potensial yang sedang menunggu untuk di eksekusi
3. Solusi: calon aksi yang akan dibandingkan

f. Fasilitas Penjelasan Sistem

Adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Digunakan untuk melacak respon dan memberi penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:

1. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
2. Bagaimana konklusi dicapai?
3. Mengapa ada alternative yang dibatalkan?
4. Rencana apa yang digunakan untuk mendapatkan solusi?
5. Perbaikan Pengetahuan

2.3 Diagnosis

Diagnosis adalah klasifikasi seseorang berdasarkan suatu penyakit yang dideritanya atau satu abnormalitas yang diidapnya. Batasan mengenai diagnosis dalam ICD-10 adalah diagnosis berarti, penyakit, cedera, cacat, keadaan masalah terkait kesehatan. Diagnosis utama adalah kondisi yang setelah pemeriksaan ternyata penyebab utama *admission* pasien ke rumah sakit untuk dirawat. Diagnosis sekunder adalah masalah kesehatan yang muncul pada saat episode keperawatan kesehatan, yang mana kondisi itu belum ada di pasien. Setiap diagnosis harus mengandung kekhususan dan etiologi. Apabila dokter tidak dapat menemukan yang khusus atau etiologi karena hasil pemeriksaan rontgen, tes laboratorium serta

pemeriksaan lain tidak dimasukkan, maka pernyataan harus dibuat sedemikian rupa yang mampu menyatakan *symptom* dan bukan penyakitnya, diagnosis harus dijelaskan sebagai meragukan atau tidak diketahui (Huffman, 1994).

2.4 Penyakit

Penyakit adalah gangguan pada bagian tubuh, atau gangguan kesehatan (Ramli & Pamoentjak, 1990).

2.5 Pencernaan

Sistem pencernaan terdiri atas sebuah saluran-saluran panjang, saluran ini mempunyai panjang kira-kira 9 meter dimulai dari mulut, faring, esofagus, usus halus, usus besar, dan dubur (anus) (Tambayong, 2001).

2.6 Penyakit Pencernaan

Penyakit pencernaan merupakan penyakit yang menyerang organ pencernaan sehingga mengganggu kerja sistem pencernaan. Beberapa faktor yang dapat menyebabkan penyakit pencernaan antara lain makanan yang kurang baik, keseimbangan nutrisi, pola makan yang tidak teratur, dan infeksi serta kelainan pada organ pencernaan. Orang yang terkena penyakit ini akan mengalami gangguan pencernaan seperti diare, maag, dan lain-lain (Ma'rifati & Kesuma, 2018).

2.7 Pengertian Web

“*Web* adalah suatu ruang informasi dimana sumber-sumber informasi dalam bentuk halaman-halaman baik teks, gambar, suara, dan video bahkan dilengkapi juga dengan *link* untuk menghubungkan dengan halaman lain, dapat diidentifikasi

oleh pengenal global yang disebut *Uniform Resource Identifier (URL)*” (Rafi’i, 2008).

2.8 Pengertian Berbasis Web

“Berkas *Web* merupakan perangkat lunak yang dapat diakses dengan menggunakan *browser*” (Sukanto & Shalahuddin, 2013).

2.9 Metode Forward Chaining

Forward Chaining merupakan strategi yang digunakan dalam Sistem Pakar untuk mendapatkan kesimpulan/keputusan yang dimulai dengan menelusuri fakta-fakta dan tempat (Ariawan, Sanjaya dan Divayana, 2016).

Forward Chaining adalah pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (*IF* dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. Pada sistem alur maju, fakta-fakta dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbaharui. Keluaran dalam sistem mempresentasikan aksi-aksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja atau sering disebut kondisi aksi. Kondisi biasanya berupa penambahan atau penghapusan item dalam memori kerja. Berikut contoh *inferensi* dengan menggunakan metode alur maju, yaitu :

JIKA demam tinggi dan bintik-bintik merah

MAKA penderita terkena penyakit demam berdarah

2.10 Basis Data

“Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya ” (Fathansyah, 2015).

2.10.1 *My Structured Query Language (MySQL)*

“*MySQL* adalah *software* atau program *Database Server*”. Sedangkan *SQL* adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam *MySQL* itu sendiri. *SQL* juga dipakai dalam *software database server* lain, seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya (Nugroho, 2013).

“*MySQL* Merupakan database server yang paling sering digunakan dalam pemrograman PHP. *MySQL* digunakan untuk menyimpan data dalam database dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam database” (Buana, 2014).

2.10.2 *Structured Query Language (SQL)*

“*SQL* (dibaca “ess-que-el”) merupakan kependekkan dari *Structured Query Language*. *SQL* digunakan untuk berkomunikasi dengan sebuah Database. Sesuai ANSI, *SQL* merupakan bahasa standar untuk sistem manajemen database relasional” (Prasetio, 2012).

2.11 Bahasa Pemrograman

2.11.1 *Hypertext Preprocessor (PHP)*

”*PHP (PHP: Hypertext Preprocessor)* adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam *HTML*” (Supono & Putratama, 2016).

“*PHP* merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web*” (Solichin, 2016).

“*PHP* merupakan bahasa (*script*) pemrograman yang sering digunakan pada sisi server sebuah *web*” (Komputer, 2010).

Kumpulan kutipan diatas menerangkan bahwa *hypertext preprocessor (PHP)* merupakan bahasa pemrograman untuk membuat/mengembangkan aplikasi berbasis *web* dan bersifat *open source* dan ditanamkan ke dalam *script HTML*.

2.11.2 *Hypertext Markup language (HTML)*

HTML adalah suatu bahasa yang dikendalikan oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*).

HTML adalah bahasa markup untuk menyebarkan informasi pada *web*. Ketika merancang *HTML*, ide ini diambil dari *Standart Generalized Markup Language (SGML)*. *SGML* adalah cara yang terstandarisasi dari pengorganisasian dan informasi yang terstruktur di dalam dokumen atau sekumpulan dokumen.

Walaupun *HTML* tidak dengan mudah dapat dipahami kebanyakan orang, ketika diterbitkan penggunaannya menjadi jelas (Oktavian, 2013).

2.11.3 *Java Script*

“*Java Script* adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada *web browser*” (Sibero, 2013).

“*Java Script* merupakan bahasa pemrograman *web* yang berbasis *script*. Menggunakan *java script* dalam pemrograman *web* dapat menciptakan sebuah *web* dengan tampilan dan kinerja yang dinamis dan interaktif” (Irawan, 2012).

2.11.4 *Cascading Style Sheet (CSS)*

“*CSS* atau yang memiliki kepanjangan *Cascading Style Sheet* merupakan suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur dan seragam” (Saputra, 2012).

“*Cascading Style Sheet (CSS)* adalah dikembangkan untuk menata gaya pengaturan halaman *web*, terdiri dari *selector* properti dan nilai seperti halnya *HTML* atau *PHP* dan bahasa pemrograman lainnya” (Sibero, 2013).

2.12 Aplikasi Pendukung

2.12.1 *PhpMyAdmin*

“*PhpMyAdmin* adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator *MySQL* melalui *browser (web)* yang digunakan

untuk *managemen database*. *PhpMyAdmin* mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, table, relasi antar table, dan lain sebagainya” (Rahman, 2013).

2.12.2 *Xampp*

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

Adapun pengertian dari *XAMPP* adalah sebagai berikut:

1. X yang berarti program ini dapat dijalankan di banyak sistem operasi
2. A yang berarti *Apache* merupakan suatu aplikasi *web server*
3. M yang berarti *MySQL* merupakan untuk penyimpanan *database server*
4. P yang berarti *PHP* merupakan bahasa pemrograman yang dipakai untuk membangun *website* dinamis.
5. P yang berarti *Perl* bahasa pemrograman untuk segala keperluan. *Perl* merupakan penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum, *perl* sangat populer digunakan dalam pemrograman CGI (*Common Gateway Interface*).

Bagian penting dari *XAMPP* yang biasa digunakan :

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

2.13 Enterprise Relationship Diagram (ERD)

“*Diagram Entity Relationship* merupakan suatu simbol/notasi dasar yang digunakan diagram E-R, yaitu entitas, relasi, atribut dan garis penghubung” (Priyadi, 2014).

a. Komponen ERD

Penjelasan untuk notasi (Priyadi, 2014), sebagai berikut:

1. Entitas merupakan notasi untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut, sehingga pada suatu lingkungan yang nyata setiap objek akan berbeda dengan objek lainnya. Pada umumnya, objek dapat berupa benda, pekerjaan, tempat, dan orang.
2. Relasi merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.
3. Atribut merupakan notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai *key* yang bersifat unik, yaitu

Primary Key atau *Foreign Key*. Selain itu, atribut juga dapat sebagai atribut deskriptif saja yaitu sebagai pelengkap deskriptif suatu entitas dan relasi.

4. Garis penghubung merupakan notasi untuk merangkaikan keterkaitan antara notasi-notasi yang digunakan dalam Diagram E-R, yaitu entitas, relasi, dan atribut.

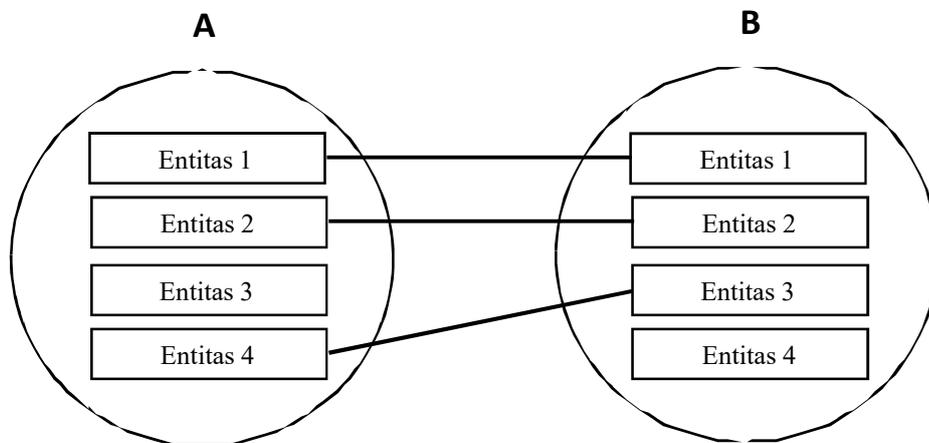
b. Kardinalitas/Derajat Relasi

“Kardinalitas Relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”. Kardinalitas Relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya (Fathansyah, 2015).

Kardinalitas Relasi yang terjadi di antara dua himpunan entitas (misalnya A dan B) dapat berupa (Fathansyah, 2015) :

1. Satu ke satu (*One to One*)

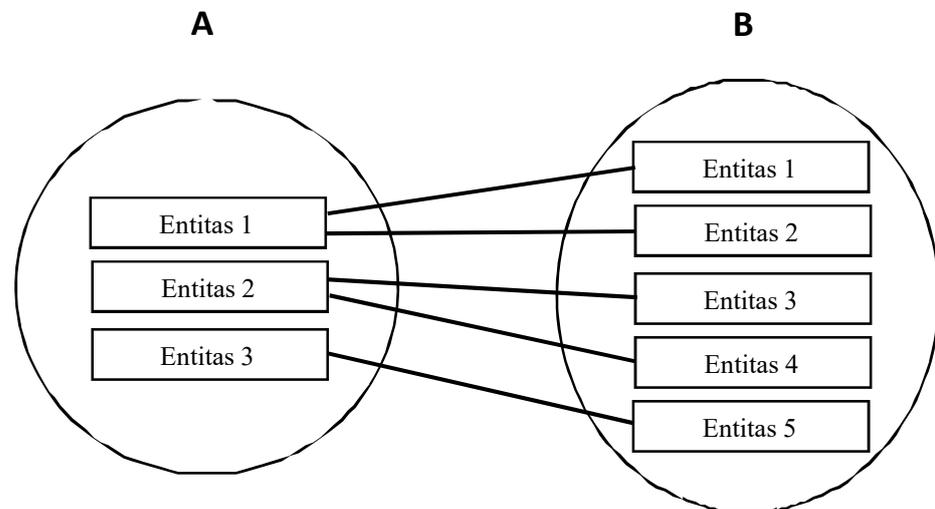
Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.2. Kardinalitas Relasi Satu ke Satu (*One to One*)

2. Satu ke Banyak (*One to Many*)

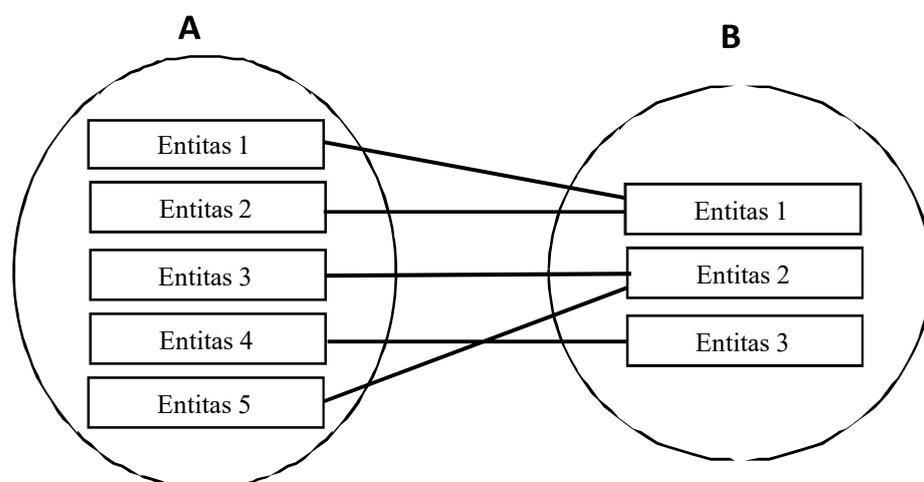
Yang berarti setiap entitas himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, di mana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.3. Kardinalitas Relasi Satu ke Banyak (One to Many)

3. Banyak ke Satu (*Many to One*)

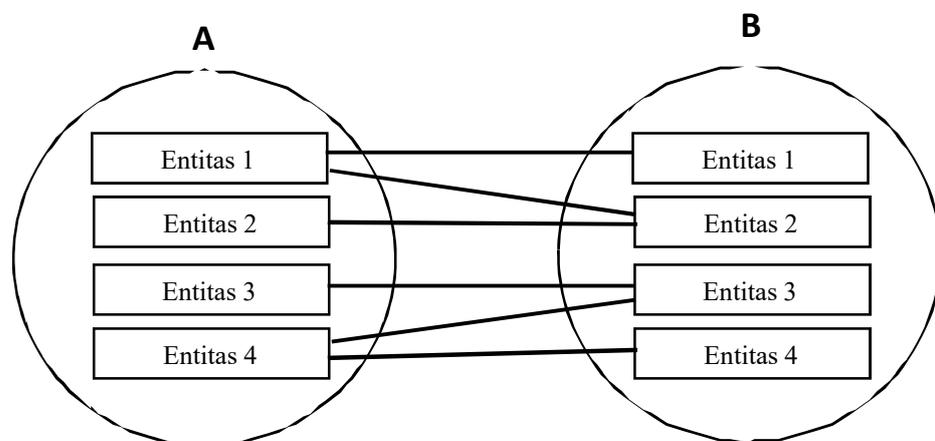
Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas.



Gambar 2.4. Kardinalitas Relasi Banyak ke Satu (Many to One)

4. Banyak ke Banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Gambar 2.5. Kardinalitas Relasi Banyak ke Banyak (*Many to Many*)

c. *Logical Record Structure (LRS)*

Membentuk skema basis data atau LRS berdasarkan diagram ER, Menurut (Simarmata, 2007), yaitu:

1. Jika relasinya satu-ke-satu, maka *foreign key* diletakkan pada salah satu dari 2 entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakkan pada entitas *Many*.
3. Jika relasinya banyak-ke-banyak, maka dibuat "*file konektor*" yang berisi 2 *foreign key* yang berasal dari kedua entitas.

2.14 *Flowchart*

Flowchart mendeskripsikan detail sebuah proses, tahapan dan urutannya secara grafis. *Flowchart* berisi bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* dapat didefinisikan sebagai sebuah gambaran yang menjelaskan proses yang akan dilihat atau dikaji. Selain itu, *flowchart* biasanya digunakan untuk merencanakan tahapan suatu kegiatan. Jadi, *Flowchart* atau bagan alur merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah (*prosedur*) beserta liran data dengan simbol-simbol standar yang mudah dipahami (Astuti, 2016). *Flowchart* yang baik setidaknya mencakup hal-hal berikut:

- a. Menggambarkan seluruh tahapan proses
- b. Dapat menjelaskan dan membantu pemecahan masalah
- c. Mengidentifikasi titik proses yang kritis untuk pengendalian
- d. Menggambarkan peluang perbaikan

2.14.1 Jenis *Flowchart*

Flowchart dapat digunakan untuk segala sesuatu yang mengandung rangkaian kegiatan dan aliran data, bahkan untuk kepentingan pribadi. *Flowchart* dapat dibedakan ke dalam tiga kategori, yaitu *flowchart* dokumen, *flowchart* sistem dan *flowchart* program. *Flowchart* dokumen menggambarkan aliran data dan informasi antar- area (divisi) dalam sebuah organisasi. Sedangkan *flowchart* sistem lebih menekankan pada gambaran tentang aliran input, prosedur pemrosesan dan output yang dihasilkan. Dan *flowchart* program lebih banyak digunakan untuk

menggambarkan logika suatu prosedur penyelesaian masalah dalam pemrograman komputer .

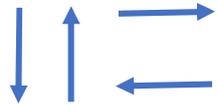
2.14.2 Simbol *Flowchart*

Flowchart memiliki simbol tersendiri yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alur dan kegunaan dari simbol-simbol yang bersangkutan dijelaskan pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol *Flowchart*

NO	SYMBOL	NAMA	FUNGSI
1		Terminal	Menyatakan permulaan atau akhir suatu program.
2		<i>Input / Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
3		<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh computer
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya atau tidak

Lanjutan Tabel 2.3 Simbol-Simbol *Flowchart*

NO	SYMBOL	NAMA	FUNGSI
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
7		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk
8		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9		<i>Punch Tape</i>	Menyatakan input atau output yang menggunakan pita kertas berlubang
10		<i>Document</i>	Menyatak keluaran dalam bentuk dokumen (melalui printer)
11		<i>Flow</i>	Menyatakan jalanna arus suatu proses

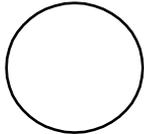
2.15 *Data Flow Diagram (DFD)*

“*Data Flow Diagram* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah refresentasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari

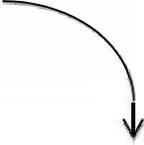
masukan (*input*) dan keluaran (*output*). *DFD* tidak sesuai untuk memodelkan sistem yang menggunakan pemograman berorientasi objek” (Sukanto & Shalahuddin, 2014).

Notasi-notasi pada *DFD* adalah sebagai berikut (Sukanto & Shalahuddin, 2014):

Tabel 2.4 Simbol-simbol *Data Flow Diagram*

NOTASI	KETERANGAN
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program</p> <p>Catatan:</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel</p>

Lanjutan Tabel 2.4 Simbol-simbol *Data Flow Diagram*

NOTASI	KETERANGAN
	<p>basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CMD)</i>, <i>Physical Data Model (PDM)</i>)</p> <p>Catatan:</p> <p>Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>Catatan:</p> <p>Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan:</p> <p>Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa”</p>

Lanjutan Tabel 2.4 Simbol-simbol *Data Flow Diagram*

NOTASI	KETERANGAN
	atau tanpa kata data misalnya “siswa”

Berikut ini adalah tahapan- tahapan perancangan dengan menggunakan *DFD* (Sukamto & Shalahuddin, 2014):

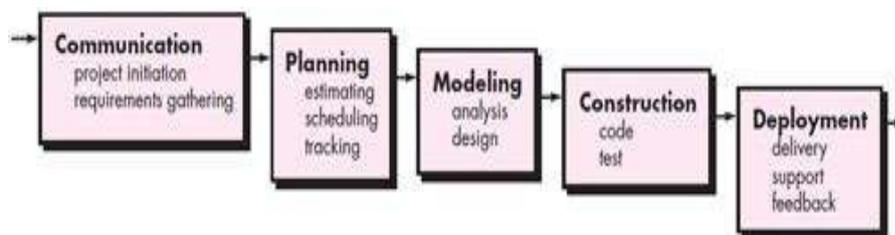
- a. Membuat *DFD Level 0* atau sering disebut juga *Context Diagram*
DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. *DFD Level 0* digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.
- b. Membuat *DFD Level 1*
DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. *DFD Level 1* merupakan hasil *breakdown DFD Level 0* yang sebelumnya sudah dibuat.
- c. Membuat *DFD Level 2*
Modul-modul pada *DFD Level 1* dapat di *breakdown* menjadi *DFD Level 2*. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingkat kedetilan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di *breakdown* lagi. Untuk sebuah sistem, jumlah *DFD Level 2* sama dengan jumlah modul pada *DFD Level 1* yang di *breakdown*.

- d. Membuat *DFD Level 3* dan seterusnya.

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada *DFD Level* di atasnya. *Breakdown* pada *level 3, 4* dan *5* dan seterusnya aturannya sama persis dengan *DFD Level 1* atau *Level 2*.

2.16 Metode Pengembangan

Metode perancangan yang digunakan dalam tugas akhir ini adalah menggunakan salah satu dari metode pengembangan perangkat lunak *software development life cycle (SDLC)* yaitu metode *waterfall* yang memiliki beberapa tahapan yaitu seperti pada gambar 2.6.



Gambar 2.6. Waterfall Pressman (Pressman, 2015)

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

1. *Communication*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin

dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. *Planning*

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

3. *Modeling*

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

4. *Construction*

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

5. *Deployment*

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

BAB III

ANALISIS MASALAH DAN PERANCANGAN PROGRAM

3.1 *Communication*

Merupakan proses-proses pengumpulan informasi atau data yang berkaitan dengan penelitian.

3.1.1 Survei

Merupakan penelitian kuantitatif menggunakan pertanyaan-pertanyaan dalam bentuk kuisioner yang akan diberikan kepada 10 responden untuk mengukur validasi data-data yang didapat.

3.1.2 Studi Literatur

Merupakan tahap dimana informasi yang diperoleh peneliti bersumber dari pengumpulan data-data yang didapat dari berbagai sumber seperti, buku, jurnal, ataupun literature yang berhubungan dengan materi sistem pakar menggunakan metode *Forward Chaining*. Berikut adalah tabel literatur yang menjadi referensi penelitian :

Tabel 3.1 Referensi Penelitian

No	<i>Literature</i>	Pembahasan
1	Imam Soleh Ma'rifati, Chandra Kesuma “Pengembangan Sistem Pakar Mendeteksi Penyakit Pencernaan Menggunakan Metode <i>Naïve Bayes</i> Berbasis Web” Jurnal Evolusi Volume 6 No 1 – 2018	Penelitian ini dibuat untuk mendeteksi penyakit pencernaan menggunakan metode <i>Naïve Bayes</i>
2	Esti Rahmawati, Hari Wibawanto “Sistem Pakar Diagnosis Penyakit Paru - Paru Menggunakan Metode <i>Forward Chaining</i> ” Jurnal Teknik Elektro Vol. 8 No. 2, Juli – Desember 2016	Penelitian ini dibuat untuk mendiagnosa penyakit paru - paru menggunakan metode <i>Forward Chaining</i> .
3	Bagus Fery Yanto, Indah Werdiningsih, Endah Purwanti “Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode <i>Forward Chaining</i> ” <i>Journal of Information Systems Engineering and Business Intelligence</i> Vol. 3, No. 1, April 2017	Penelitian ini dibuat untuk mendiagnosa penyakit pada anak bawah lima tahun menggunakan metode <i>Forward Chaining</i> .

Lanjutan Tabel 3.1 Referensi Penelitian

No	<i>Literature</i>	Pembahasan
4	Mohamad Hadi, M. Misdrum, Ratih Fitri Aini “Perancangan Sistem Pakar Diagnosa Penyakit Ayam Dengan Metode <i>Forward Chaining</i> ” JIMP – Jurnal Informatika Merdeka Pasuruan Vol. 2, No. 1 Agustus 2016	Penelitian ini dibuat untuk mendiagnosa penyakit ayam menggunakan metode <i>Forward Chaining</i>
5	Ahmad Aniq Noor Mustaqof, Wiharto S.T M.Kom, Esti Suryani S.Si M.Kom “Sistem Pakar Untuk Mendiagnosis Penyakit Infeksi Menggunakan Metode <i>Forward Chaining</i> ” Jurnal ITSMART Vol 4. No 1. Juni 2015	Penelitian ini dibuat untuk mendiagnosa penyakit infeksi menggunakan metode <i>Forward Chaining.</i>
6	Afriosa Syawitri, Sarjon Defit, Gunadi Widi Nurcahyo “Diagnosis Penyakit Gigi dan Mulut Dengan Metode <i>Forward Chaining</i> ” Jurnal Sains, Teknologi dan Industri, Vol. 16, No. 1, Desember 2018, pp. 24 – 29	Penelitian ini dibuat untuk mendiagnosa penyakit gigi dan mulut menggunakan metode <i>Forward Chaining.</i>

Kesimpulan yang dapat diambil berdasarkan literatur diatas yaitu dari segi keakuratannya rata-rata mencapai angka 80%-90% dalam menggunakan metode *forward chaining*.

3.2 *Planning*

3.2.1 Penjadwalan

Pada tahap ini penelitian mengutamakan pada penjadwalan pengerjaan penelitian. Terdapat beberapa proses yang harus dilakukan dari tahap *communication* sampai *implementation* dan *testing* maka diperlukannya penjadwalan yang tepat agar penelitian ini selesai pada waktunya. Berikut adalah penjadwalan penelitian berdasarkan aktifitas yang dilakukan dengan skala waktu.

Tabel 3.2. Penjadwalan Penelitian

No	Tahap	Aktivitas	November				Desember				Januari	
			1	2	3	4	1	2	3	4	1	2
1	Communication	Metode Pengumpulan Data										
2		Studi Literatur										
3	Planning	Penjadwalan										
4	Modelling	Analisis Sistem										
5		Perancangan Sistem										
6	Construction	Implementasi <i>Hardware & Software</i>										
7		Testing										

3.3 Modelling

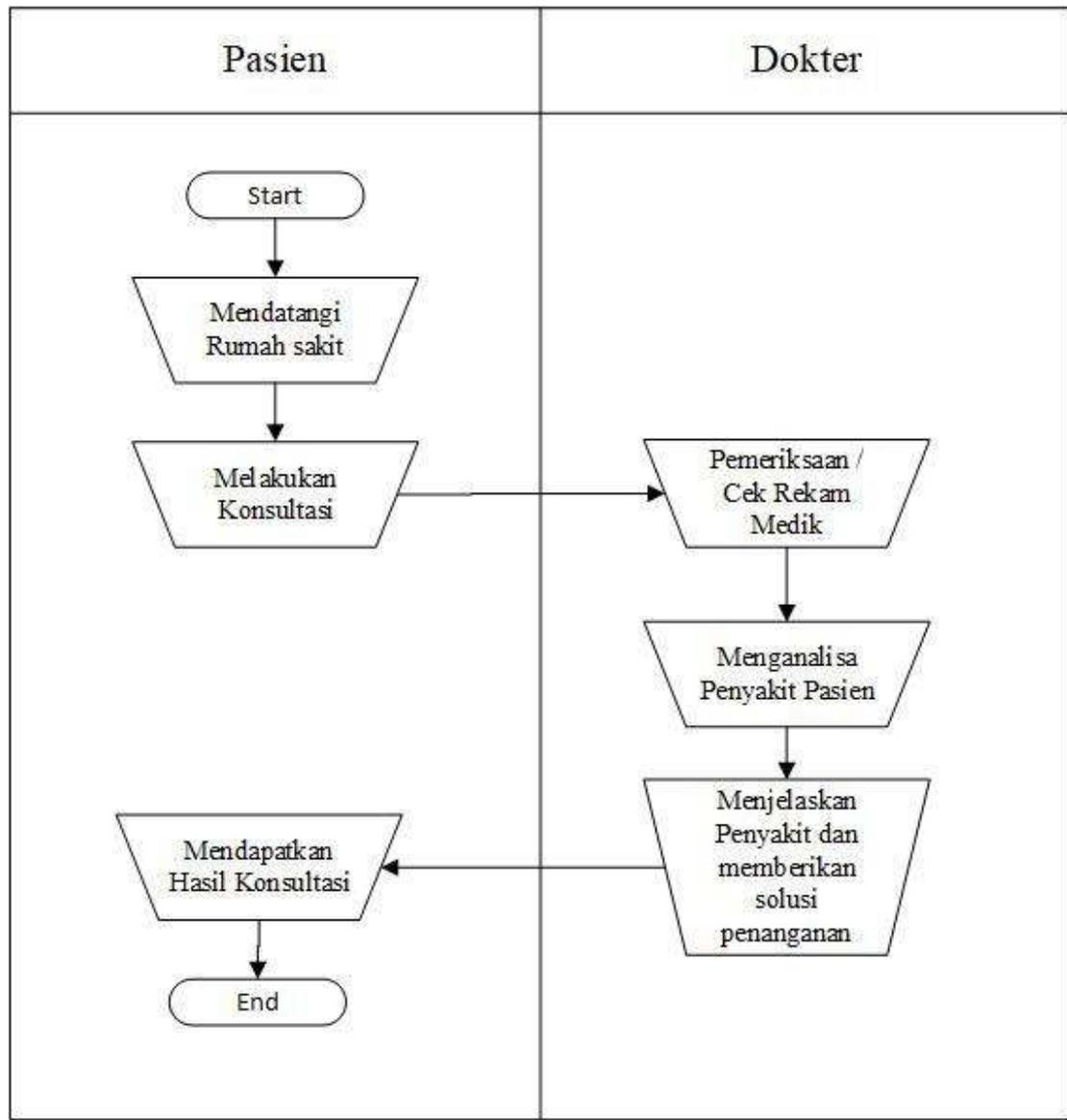
3.3.1 Analisis Sistem

3.3.1.1 Analisis Sistem yang Sedang Berjalan

Sistem yang sedang berjalan dimasyarakat pada diagnosis penyakit pencernaan adalah sebagai berikut :

- a. Pasien yang akan didiagnosis harus mengunjungi rumah sakit.
- b. Pasien tersebut berkonsultasi atas gejala yang dialami kepada dokter.
- c. Setelah melakukan konsultasi, dokter akan melakukan pemeriksaan/cek rekam medik orang tersebut.
- d. Setelah melakukan pemeriksaan/cek rekam medik, dokter akan menganalisa penyakit pasien tersebut.
- e. Dokter menjelaskan penyakit apa yang di alami orang tersebut, beserta solusi untuk menangani penyakitnya.
- f. Pasien yang melakukan konsultasi akan mendapatkan hasil dari kegiatan konsultasi yang dilakukan.

Berikut adalah *flow map* dari sistem yang sedang berjalan dimasyarakat saat ini :



Gambar 3.1. Sistem yang Sedang Berjalan

3.3.1.2 Analisis Masalah

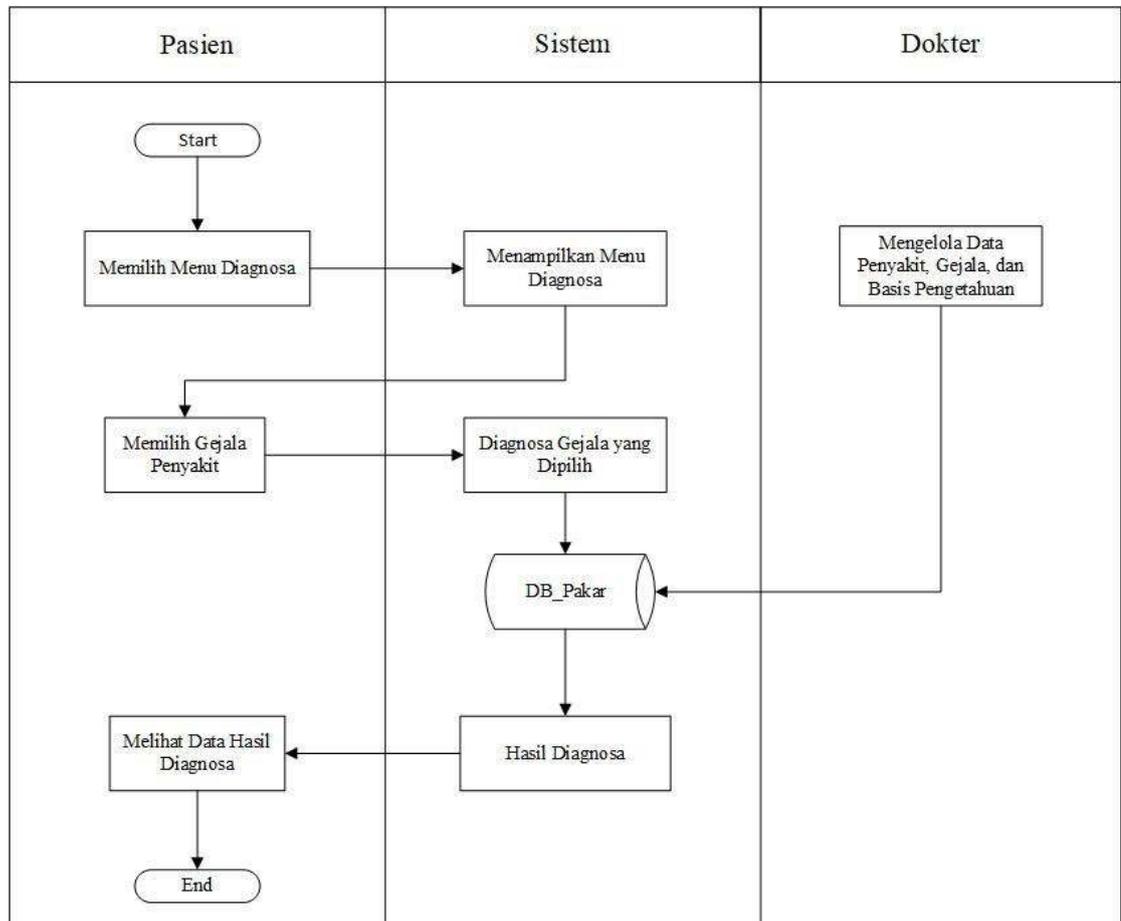
Pada Tahap ini, menjelaskan masalah yang terjadi pada sistem yang berjalan. Masalah yang terjadi yaitu, disaat pasien mengalami suatu gejala penyakit pencernaan pasien tidak dapat mengetahui penyakit yang diderita. Dibuatnya sistem

pakar ini untuk memberikan gambaran awal kepada pasien tentang penyakit yang diderita.

3.3.1.3 Analisis Sistem yang Diusulkan

Sistem yang diusulkan pada Sistem Pakar Diagnosa Penyakit Pencernaan berupa:

- a. Pasien yang akan berkonsultasi membuka halaman *website* diagnosa penyakit pencernaan dan klik pada menu diagnosa.
- b. Pasien yang akan berkonsultasi memilih gejala yang dialami pada menu diagnosa.
- c. Pasien yang akan berkonsultasi akan menerima informasi berupa penyakit pencernaan yang dialami dan solusi penanganan berdasarkan gejala yang dipilih pada menu diagnosa.



Gambar 3.2. Sistem yang Diusulkan

3.3.1.4 Analisis Penyakit dan Gejala Pencernaan

Sumber data yang didapat dalam penelitian ini, didapatkan dengan cara mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia, seperti melakukan wawancara dengan pakar dan studi literatur tentang materi yang berkaitan dengan penyakit pencernaan.

Tabel 3.3. Tabel Penyakit Pencernaan

Kode Penyakit	Penyakit
P01	Crohn
P02	Gastritis Akut
P03	Gastritis Kronis
P04	Maag
P05	Tukak Lambung
P06	Gastroporesis
P07	Gastroenteristis
P08	Usus Buntu
P09	Tifus
P10	Gerd
P11	Iritasi Usus Besar

Tabel 3.4. Tabel Gejala Penyakit Pencernaan

Kode Gejala Penyakit	Gejala Penyakit
G01	Mual
G02	Kembung
G03	Hilang (berkurang) nafsu makan
G04	Muntah
G05	Diare
G06	Tinja bercampur darah
G07	Nyeri perut
G08	Sesekali muntah darah
G09	Dada terasa sesak
G10	Kram perut
G11	Sesekali muntah darah
G12	Sering muntah darah
G13	Perih dari leher sampai perut
G14	Nyeri dileher, pusar dan punggung
G15	Nyeri ulu hati
G16	Berat badan turun drastic

Lanjutan Tabel 3.4. Tabel Gejala Penyakit Pencernaan

Kode Gejala Penyakit	Gejala Penyakit
G17	Lidah berwarna putih
G18	Sakit saat buang air

3.3.1.5 Analisis Tabel Basis Pengetahuan

Tabel basis pengetahuan digunakan untuk acuan dalam pembuatan pohon keputusan dan kaidah yang digunakan.

Tabel 3.5. Tabel Basis Pengetahuan

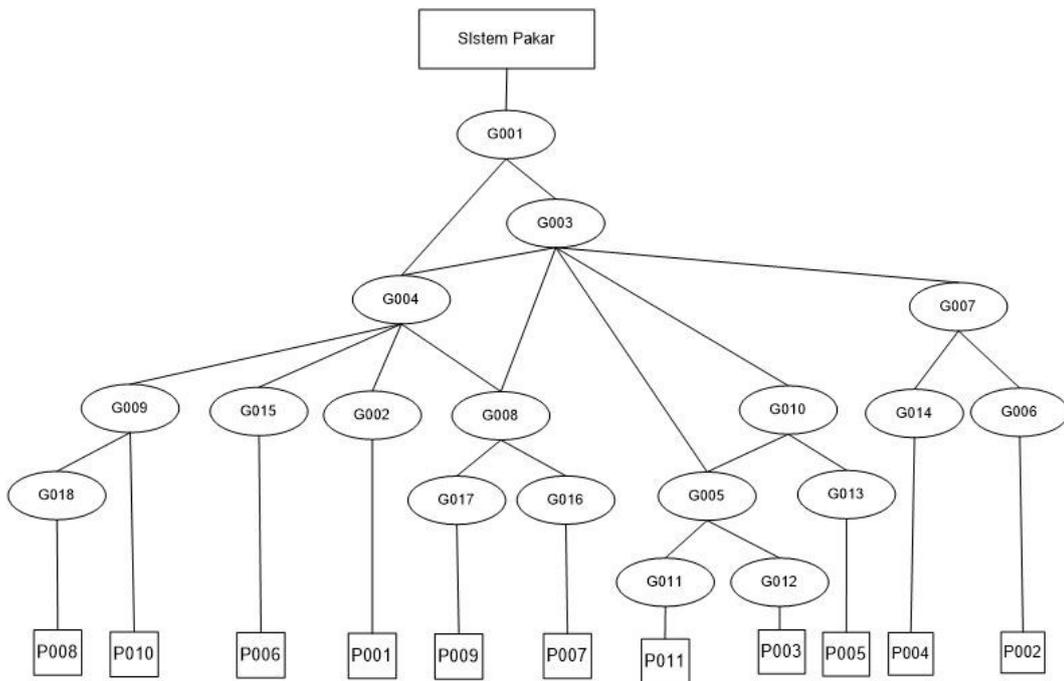
ID	Penyakit										
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11
G01	X	X	X		X	X	X	X	X	X	X
G02	X										
G03	X	X	X	X	X	X	X	X			
G04	X					X			X	X	
G05			X								X
G06		X									
G07		X		X							

Lanjutan Tabel 3.5. Tabel Basis Pengetahuan

ID	Penyakit										
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11
G08							X		X		
G09								X		X	
G10					X						X
G11											X
G12			X								
G13					X						
G14				X	X						
G15						X					
G16							X				
G17									X		
G18								X			

3.3.1.6 Pohon Pelacakan

Pohon pelacakan yang digunakan yaitu menggunakan metode *Best-First-Search*, dapat dilihat pada gambar berikut:



Gambar 3.3. Pohon Pelacakan

3.3.1.7 Analisis Perangkat Keras Untuk Membangun Sistem

Perangkat keras yang digunakan adalah sebagai berikut :

1. *Processor* Intel(R) Core(TM) i5 CPU @ 2.70GHz
2. *VGA* Intel(R) HD Graphics 4600
3. RAM 8192 MB
4. *Hard disk* 465 GB
5. *Monitor*
6. *Mouse*
7. *Keyboard*

3.3.1.8 Analisis Perangkat Lunak Untuk Membangun Sistem

Perangkat Lunak yang digunakan adalah sebagai berikut :

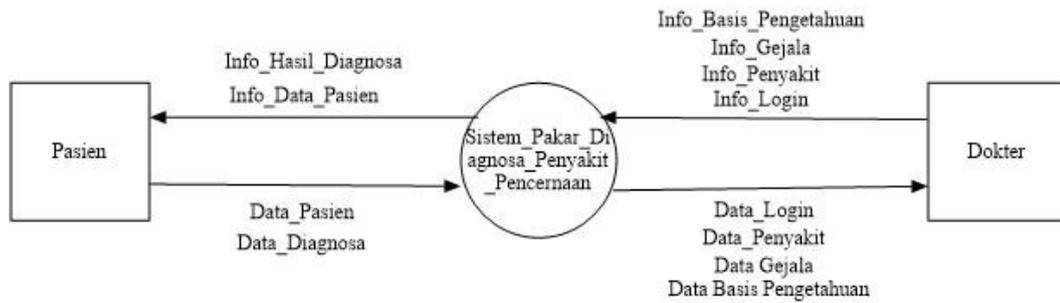
1. Sistem Operasi *Windows 10*
2. *MySQL*
3. *PHP*
4. *Jquery*
5. *Apache*
6. *Notepad++*

3.3.2 Perancangan Sistem

Perancangan sistem digunakan untuk memberikan gambaran bagaimana sistem akan berfungsi bila telah tersusun dalam bentuk yang lengkap. Perancangan sistem yang digunakan penulis berupa *Data Flow Diagram (DFD)*.

3.3.2.1 Context Diagram

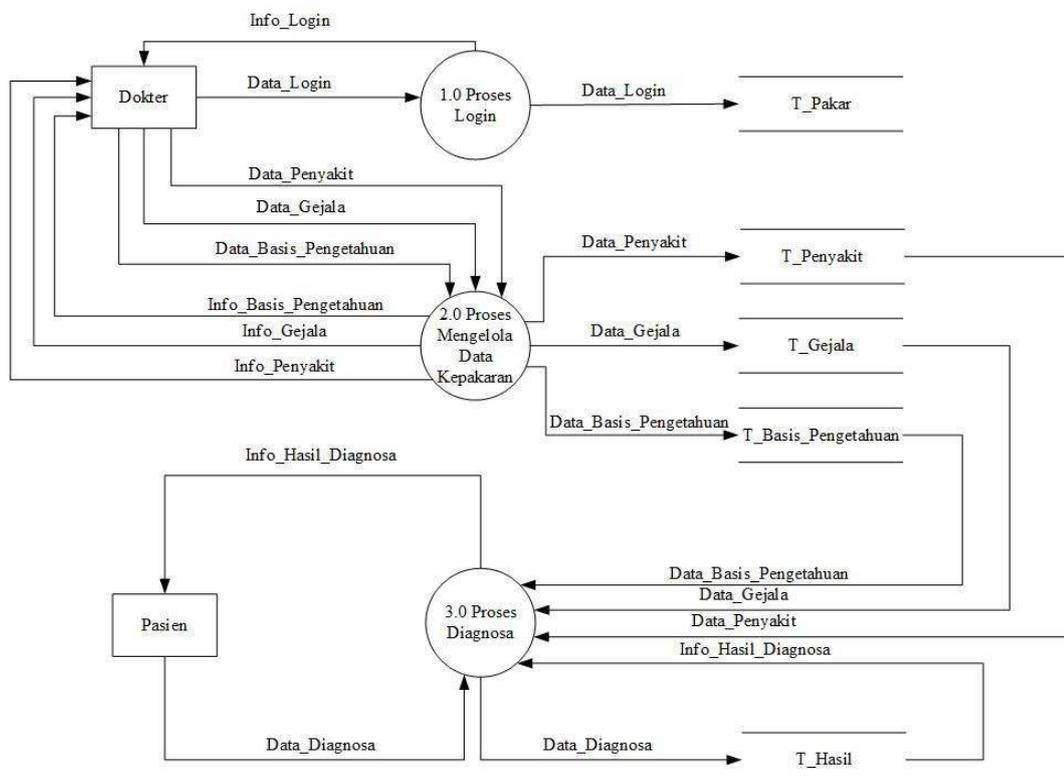
Context Diagram sistem pakar ini terdiri dari dua *entity*, yang yaitu *entity* pasien dan *entity* dokter sebagai *admin*. Berikut adalah *Context Diagram* yang dirancang oleh penulis :



Gambar 3.3. Context Diagram

3.3.2.2 DFD Level 1

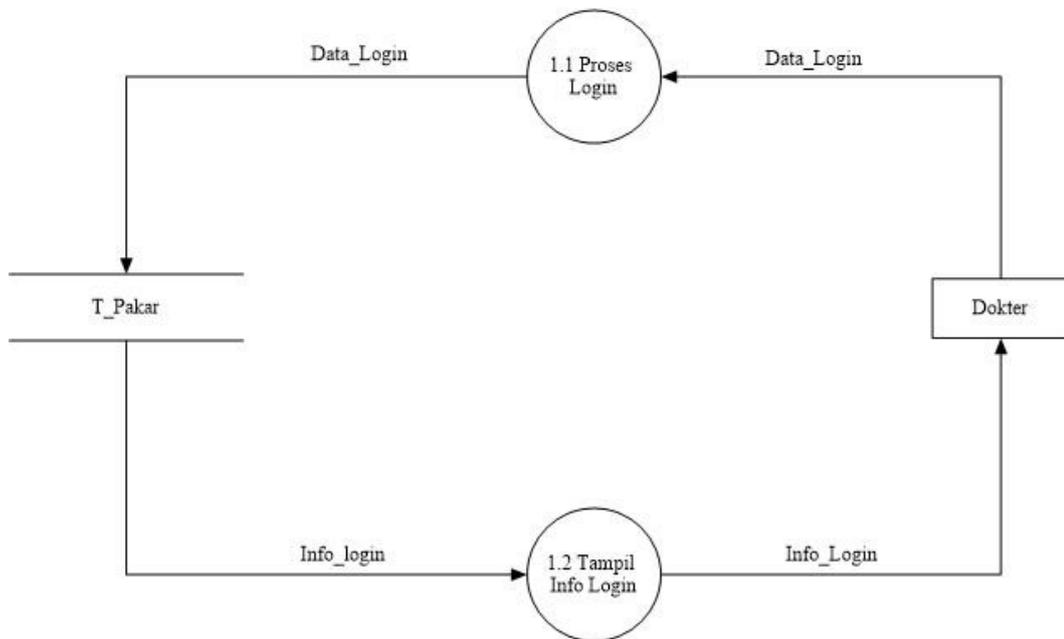
Pada diagram ini terdapat tiga macam proses, proses pertama adalah *login* untuk admin, proses kedua merupakan proses mengelola data kepakaran yang dilakukan oleh dokter sebagai *admin*, proses ketiga merupakan proses diagnosa yang dilakukan oleh pasien. Berikut ini merupakan gambar perancangan untuk DFD Level 1:



Gambar 3.4. DFD Level 1 Sistem Pakar Diagnosa Penyakit Pencernaan

3.3.2.3 DFD Level 2 Proses 1.0 (Proses Login)

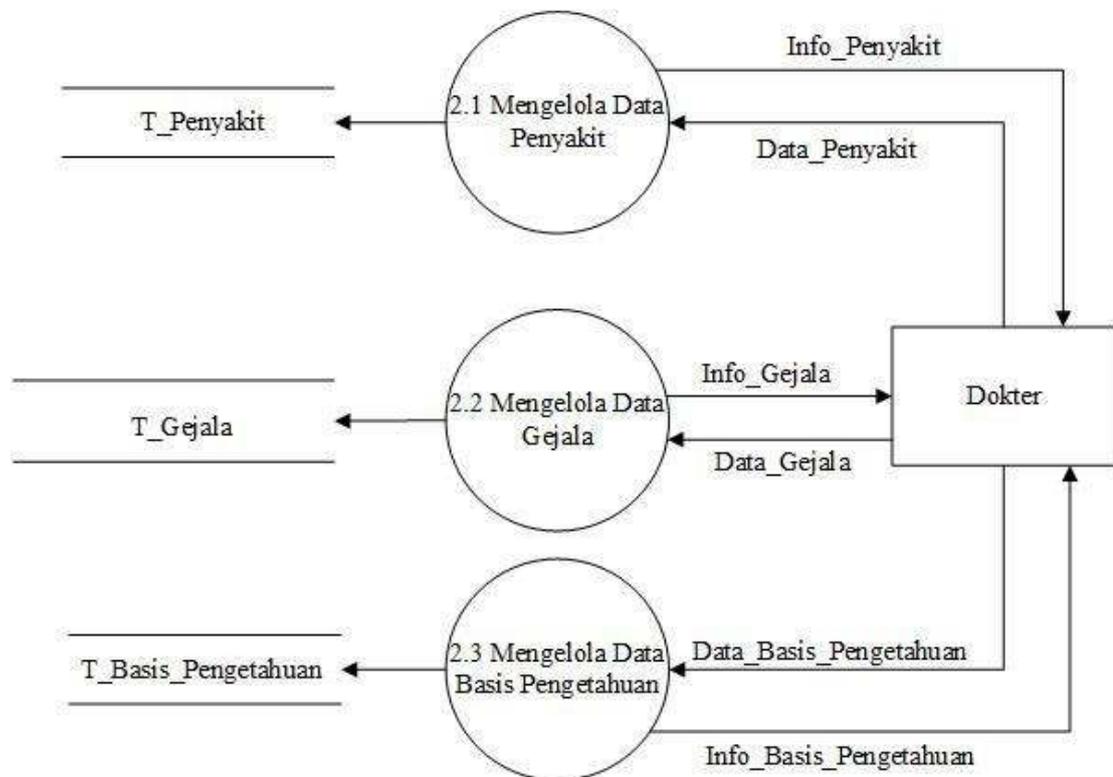
Pada diagram ini terdapat 2 (dua) macam proses. Proses pertama yaitu proses login dan proses kedua yaitu tampil info login. Berikut ini merupakan perancangan untuk DFD Level 2 Proses 1.0 (Proses login):



Gambar 3.5. DFD Level 2 Proses 1.0 (Proses Login)

3.3.2.4 DFD Level 2 Proses 2.0 (Proses Mengelola Data Kepakaran)

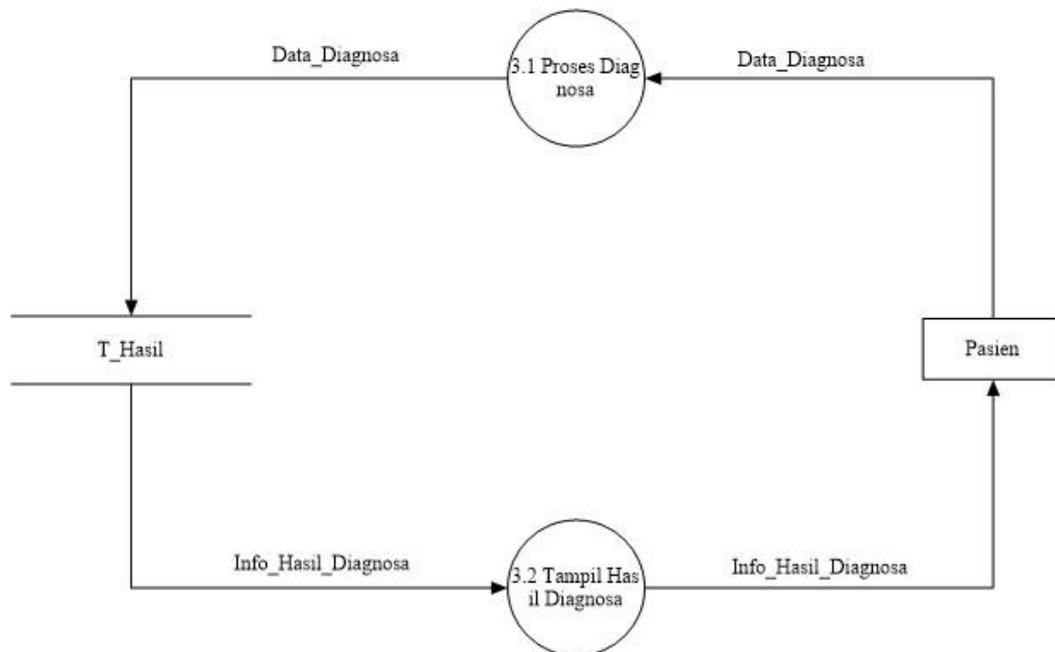
Pada diagram ini terdapat 3 (tiga) macam proses. Proses pertama yaitu proses mengelola data penyakit, proses kedua yaitu mengelola data gejala, dan proses ketiga yaitu mengelola data basis pengetahuan, Berikut ini merupakan perancangan untuk DFD Level 2 Proses 2.0 (Proses Mengelola Data Kepakaran):



Gambar 3.6. DFD Level 2 Proses 2.0 (Proses Mengelola Data Keperakan)

3.3.2.5 DFD Level 2 Proses 3.0 (Proses Diagnosa)

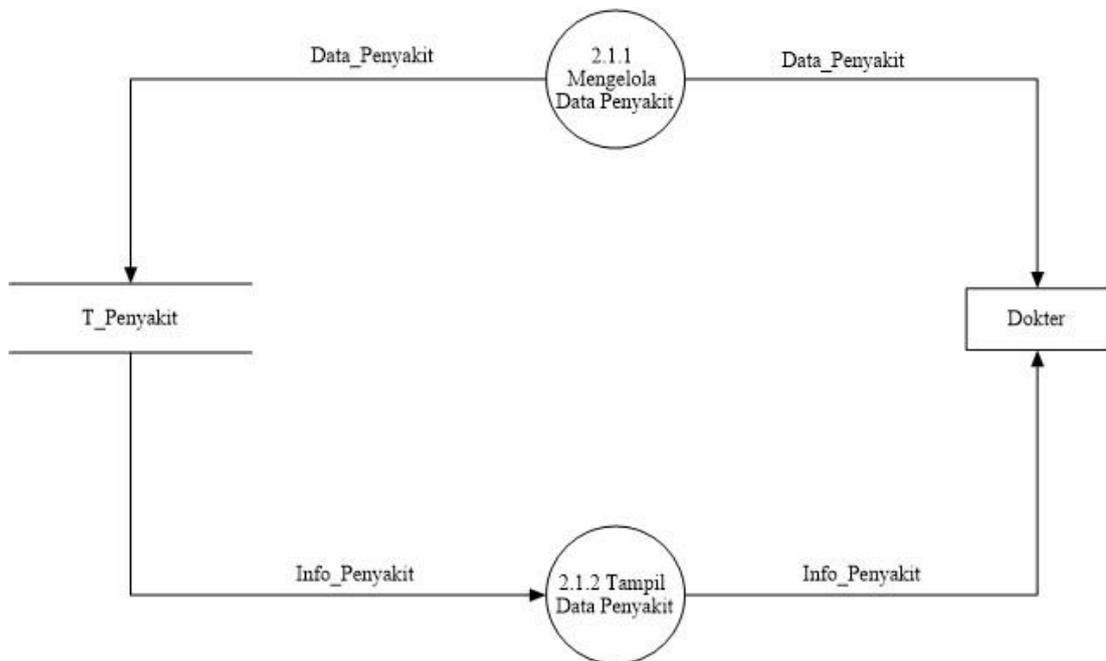
Pada diagram ini terdapat 2 (dua) macam proses. Proses pertama yaitu proses diagnosa dan proses kedua yaitu tampil hasil diagnosa. Berikut ini merupakan perancangan untuk DFD Level 2 Proses 3.0 (Proses Diagnosa):



Gambar 3.7. DFD Level 2 Proses 3.0 (Proses Diagnosa)

3.3.2.6 DFD Level 3 Proses 2.1 (Mengelola Data Penyakit)

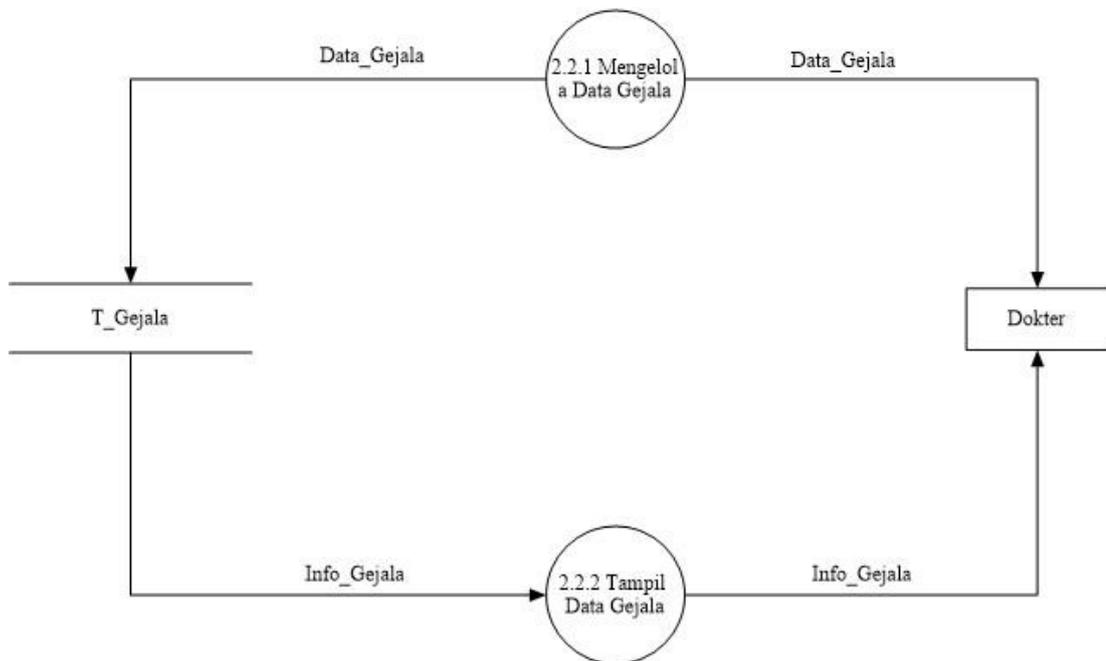
Pada diagram ini terdapat 2 (dua) macam proses. Proses pertama yaitu proses mengelola data penyakit dan proses kedua yaitu tampil data penyakit. Berikut ini merupakan perancangan untuk DFD Level 3 Proses 2.1 (Mengelola Data Penyakit):



Gambar 3.8. DFD Level 3 Proses 2.1 (Mengelola Data Penyakit)

3.3.2.7 DFD Level 3 Proses 2.2 (Mengelola Data Gejala)

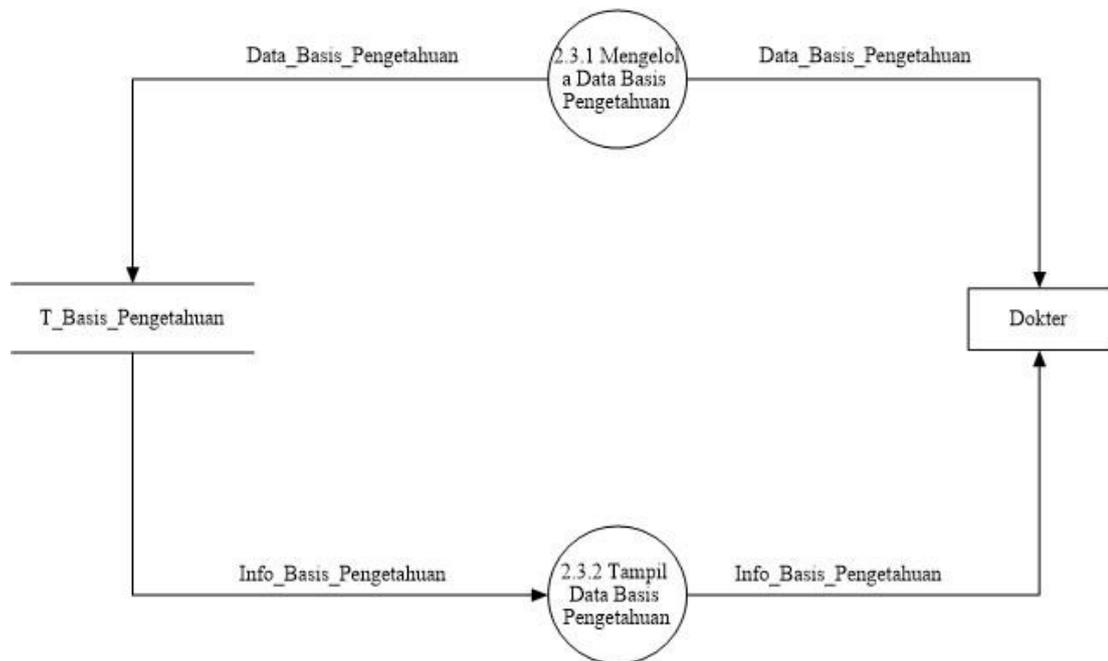
Pada diagram ini terdapat 2 (dua) macam proses. Proses pertama yaitu proses mengelola data gejala dan proses kedua yaitu tampil data gejala. Berikut ini merupakan perancangan untuk DFD Level 3 Proses 2.2 (Mengelola Data Gejala):



Gambar 3.9. DFD Level 3 Proses 2.2 (Mengelola Data Gejala)

3.3.2.8 DFD Level 3 Proses 2.3 (Mengelola Data Basis Pengetahuan)

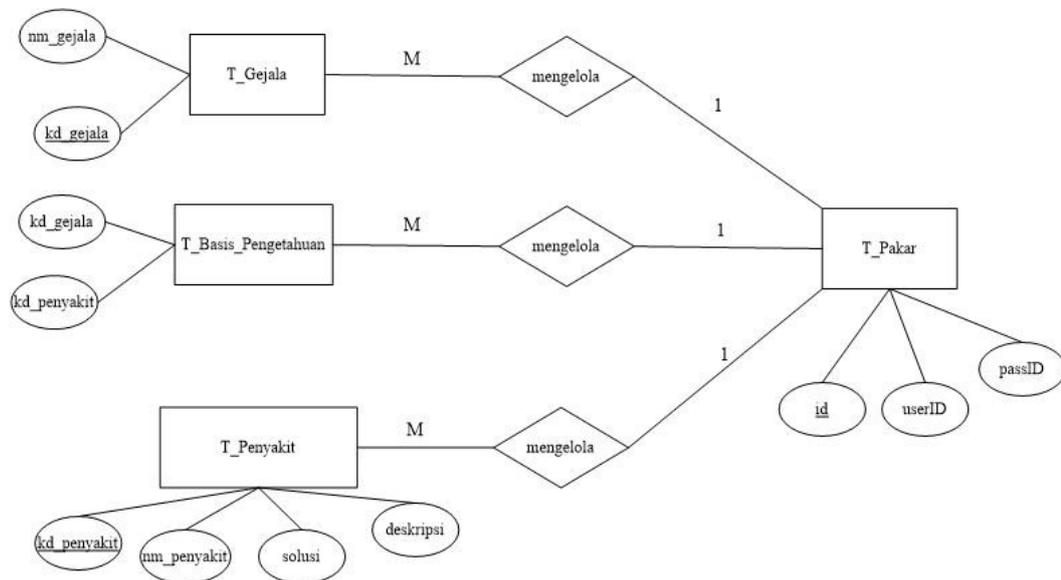
Pada diagram ini terdapat 2 (dua) macam proses. Proses pertama yaitu proses mengelola data basis pengetahuan dan proses kedua yaitu tampil data basis pengetahuan. Berikut ini merupakan perancangan untuk DFD Level 3 Proses 2.3 (Mengelola Data Basis Pengetahuan):



Gambar 3.10. DFD Level 3 Proses 2.3 (Mengelola Data Basis Pengetahuan)

3.3.2.9 ERD (Entity Relationship Diagram)

ERD atau *Entity Relationship Diagram* adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Berikut ini merupakan ERD yang dirancang :



Gambar 3.11. ERD Sistem Pakar Diagnosa Penyakit Pencernaan

Terdapat 4 (empat) tabel pada sistem pakar ini, yaitu tabel admin, tabel penyakit dan solusi, tabel gejala dan tabel basis pengetahuan. Berikut beberapa penjelasan mengenai tabel-tabel tersebut.

Tabel 3.6. Tabel Admin

Nama	Tipe Data	Ukuran	Key
Id	Int	11	<i>primary key</i>
<i>userID</i>	Varchar	255	-
<i>passID</i>	Varchar	255	-

Tabel 3.7. Tabel Penyakit

Nama	Tipe Data	Ukuran	Key
kd_penyakit	Int	20	<i>primary key</i>
nm_penyakit	Varchar	50	-
solusi	Longtext	-	-
deskripsi	Longtext	-	-

Tabel 3.8. Tabel Gejala

Nama	Tipe Data	Ukuran	Key
Id	Int	20	<i>Primary key</i>
Kode	Varchar	5	-
Gejala	Varchar	200	-

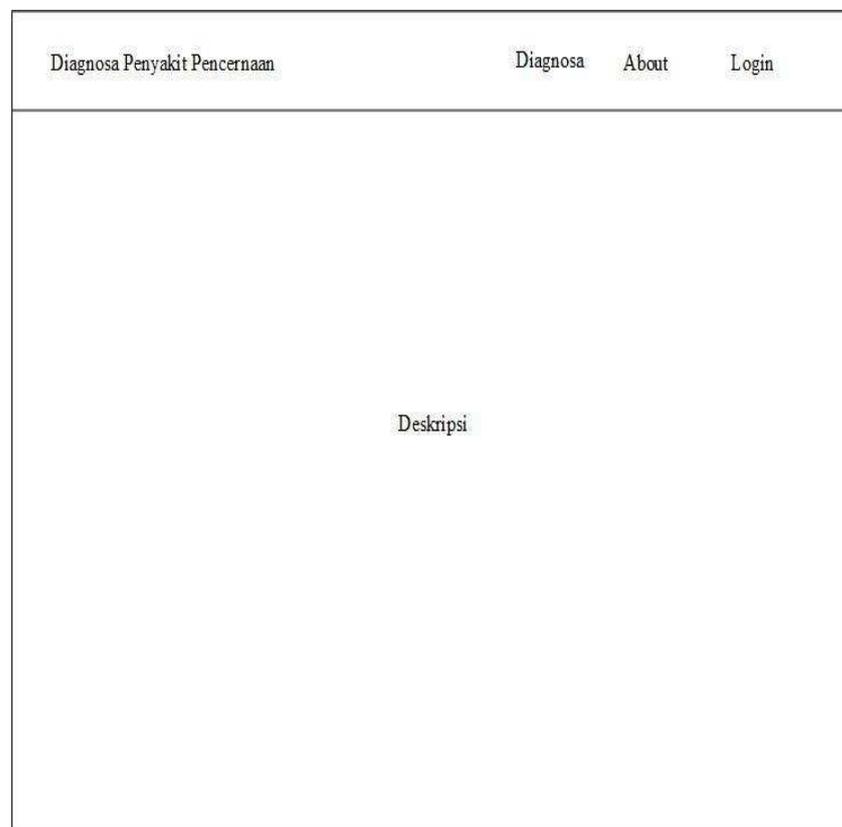
Tabel 3.9. Tabel Basis Pengetahuan

Nama	Tipe Data	Ukuran	Key
kd_gejala	Int	25	<i>primary key</i>
nm_gejala	Varchar	255	-

3.3.3 *Design Interface*

Design Interface adalah rancangan antarmuka yang digunakan sebagai perantara *user* dengan perangkat yang dikembangkan. Berikut ini merupakan beberapa *design interface* dari sistem pakar ini :

a. *Design Interface* Halaman Utama



Gambar 3.12. *Design Interface* Halaman Utama

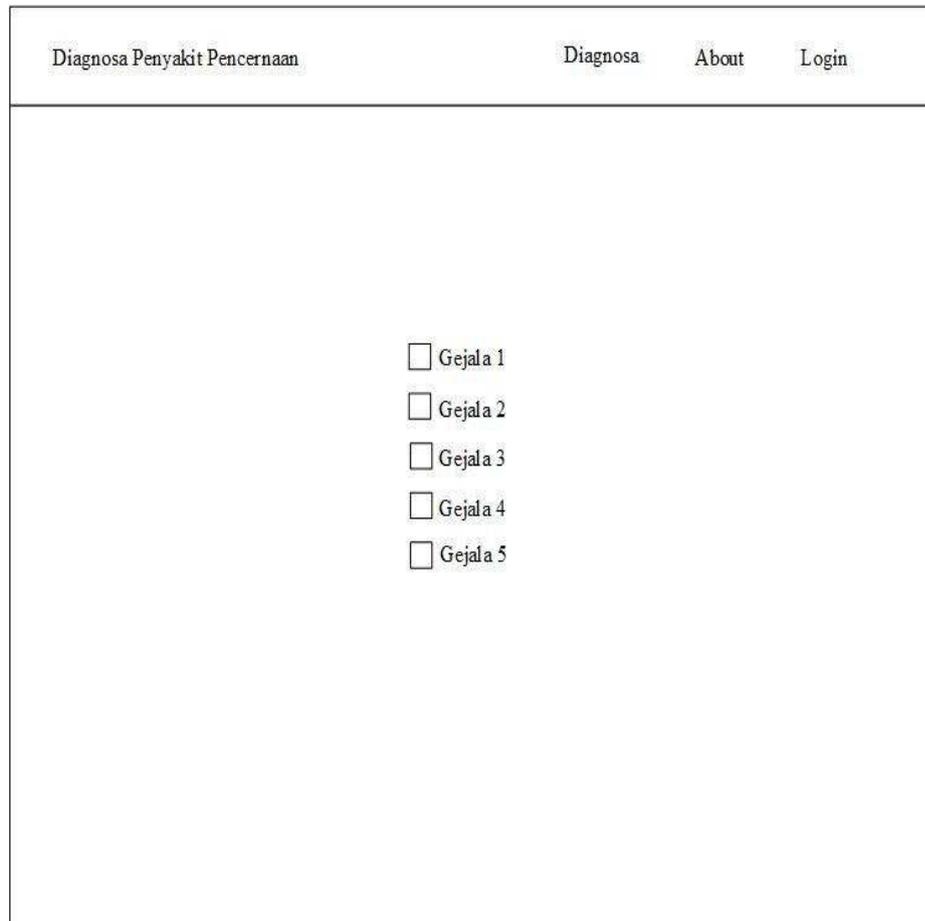
b. *Design Interface* Halaman About

No	Nama Penyakit	Deskripsi
1	Penyakit 1	Deskripsi 1
2	Penyakit 2	Deskripsi 2
3	Penyakit 3	Deskripsi 3
4	Penyakit 4	Deskripsi 4

Diagnosa Penyakit Pencernaan DIAGNOSA ABOUT LOGIN

Gambar 3.13. *Design Interface* Halaman About

c. *Design Interface* Halaman Diagnosa

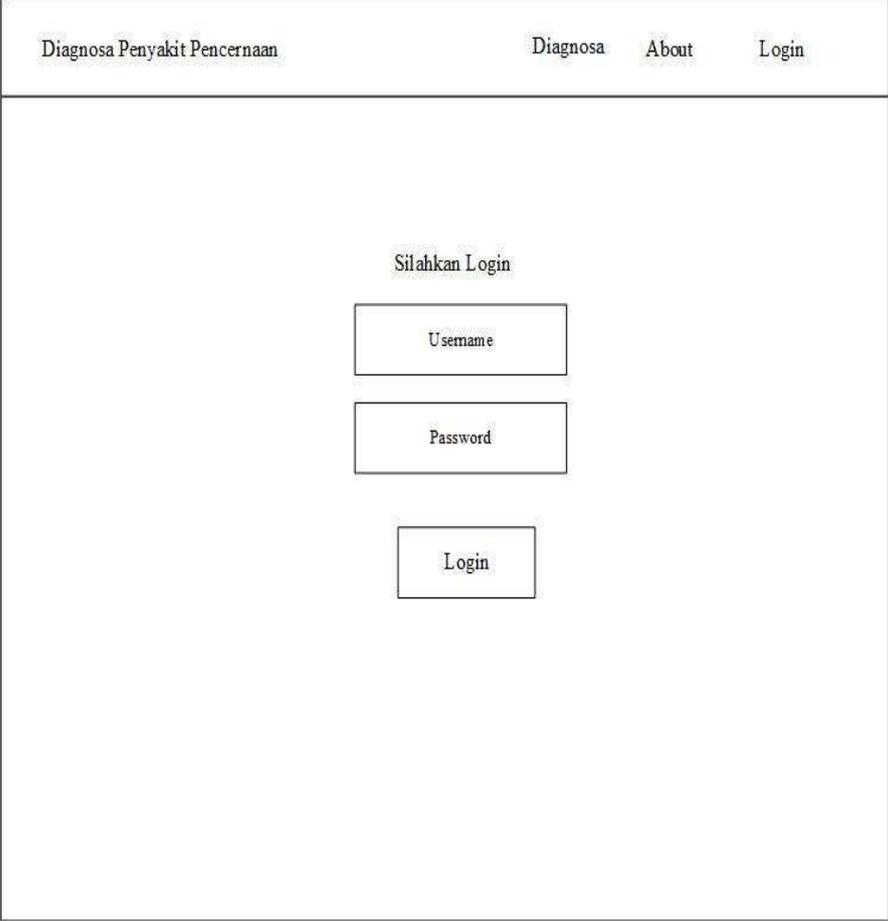


The image shows a web interface for a diagnosis page. At the top, there is a header bar with the text "Diagnosa Penyakit Pencernaan" on the left and three navigation links: "Diagnosa", "About", and "Login" on the right. Below the header, the main content area is mostly empty, with a vertical list of five symptoms in the center. Each symptom is preceded by an unchecked checkbox:

- Gejala 1
- Gejala 2
- Gejala 3
- Gejala 4
- Gejala 5

Gambar 3.14. *Design Interface* Halaman Diagnosa

d. *Design Interface Halaman Login*



The image shows a wireframe for a login page. At the top, there is a navigation bar with four items: "Diagnosa Penyakit Pencernaan" on the left, and "Diagnosa", "About", and "Login" on the right. Below the navigation bar, the main content area is centered and contains the text "Silahkan Login". Underneath this text are three vertically stacked rectangular input fields: the first is labeled "Username", the second is labeled "Password", and the third is a "Login" button.

Gambar 3.15. Design Interface Halaman Login

e. *Design Interface* Halaman Utama Admin



Gambar 3.16. *Design Interface* Halaman Utama Admin

f. *Design Interface* Halaman *Rule Admin*

Diagnosa Penyakit Pencernaan						RULE	GEJALA	PENYAKIT	LOGOUT
No	Kode Gejala 1	Kode Gejala 2	Kode Gejala 3	Kode Gejala 4	Aksi				
ID Penyakit 1	1	0	0	0	Tambah Ubah Hapus				
ID Penyakit 2	0	1	0	0	Tambah Ubah Hapus				
ID Penyakit 3	0	0	1	0	Tambah Ubah Hapus				
ID Penyakit 4	0	0	0	1	Tambah Ubah Hapus				

Gambar 3.17. *Design Interface* Halaman *Rule Admin*

g. *Design Interface* Halaman Gejala Admin

Diagnosa Penyakit Pencernaan				RULE	GEJALA	PENYAKIT	LOGOUT
Masukkan Kode Gejala		Masukkan Gejala Penyakit		Tombol Tambah Data			
No	Kode	Gejala	Aksi				
1	Kode Gejala 1	Gejala Penyakit 1	Ubah Hapus				
2	Kode Gejala 2	Gejala Penyakit 2	Ubah Hapus				
3	Kode Gejala 3	Gejala Penyakit 3	Ubah Hapus				
4	Kode Gejala 4	Gejala Penyakit 4	Ubah Hapus				

Gambar 3.18. *Design Interface* Halaman Gejala Admin

h. *Design Interface* Halaman Penyakit Admin

Diagnosa Penyakit Pencernaan	RULE	GEJALA	PENYAKIT	LOGOUT
Masukkan Nama Penyakit	Masukkan Solusi	Masukkan Deskripsi	Tombol Tambah Data	
No	Nama Penyakit	Solusi	Aksi	
1	Penyakit 1	Solusi 1	Ubah Hapus	
2	Penyakit 2	Solusi 2	Ubah Hapus	
3	Penyakit 3	Solusi 3	Ubah Hapus	
4	Penyakit 4	Solusi 4	Ubah Hapus	

Gambar 3.19. *Design Interface* Halaman Penyakit Admin

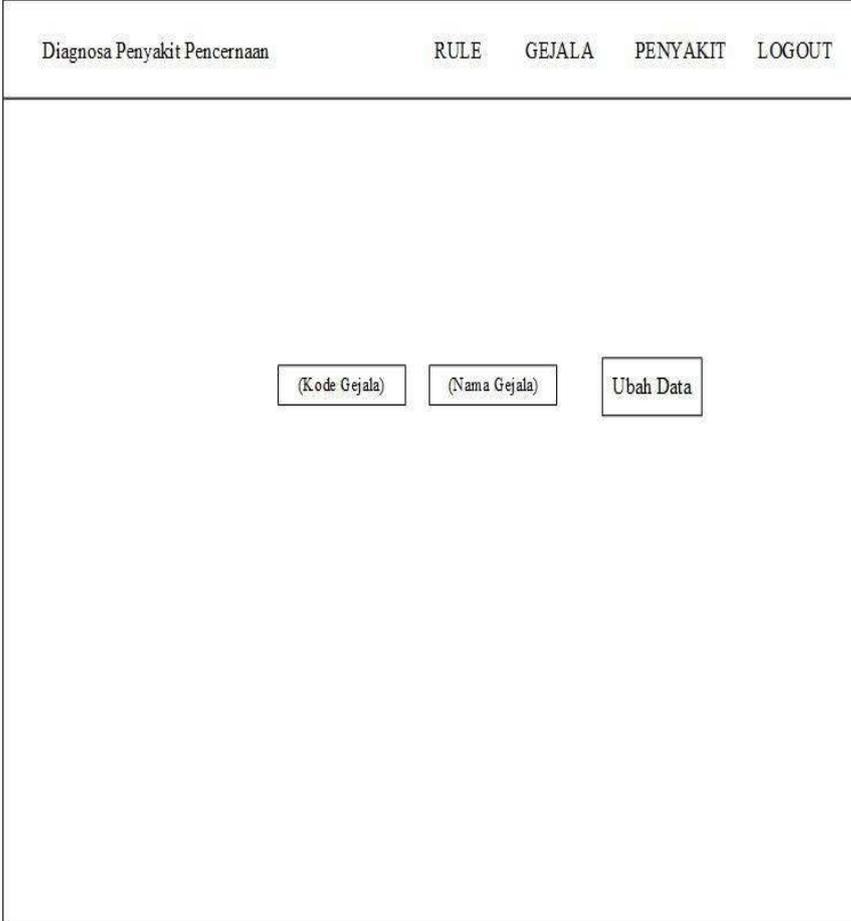
i. *Design Interface Halaman Update Rule*

Diagnosa Penyakit Pencernaan	RULE	GEJALA	PENYAKIT	LOGOUT
------------------------------	------	--------	----------	--------

Kode Gejala 1	<input type="text" value="(Data Rule)"/>
Kode Gejala 2	<input type="text" value="(Data Rule)"/>
Kode Gejala 3	<input type="text" value="(Data Rule)"/>
Kode Gejala 4	<input type="text" value="(Data Rule)"/>

Gambar 3.20. Design Interface Halaman Update Rule

j. *Design Interface Halaman Update Gejala*



The image shows a web interface for updating symptoms. At the top, there is a navigation bar with the text "Diagnosa Penyakit Pencernaan" on the left and four menu items: "RULE", "GEJALA", "PENYAKIT", and "LOGOUT" on the right. Below the navigation bar is a large white area containing three input fields and one button. The first two fields are labeled "(Kode Gejala)" and "(Nama Gejala)", and the third is a button labeled "Ubah Data".

Gambar 3.21. *Design Interface Halaman Update Gejala*

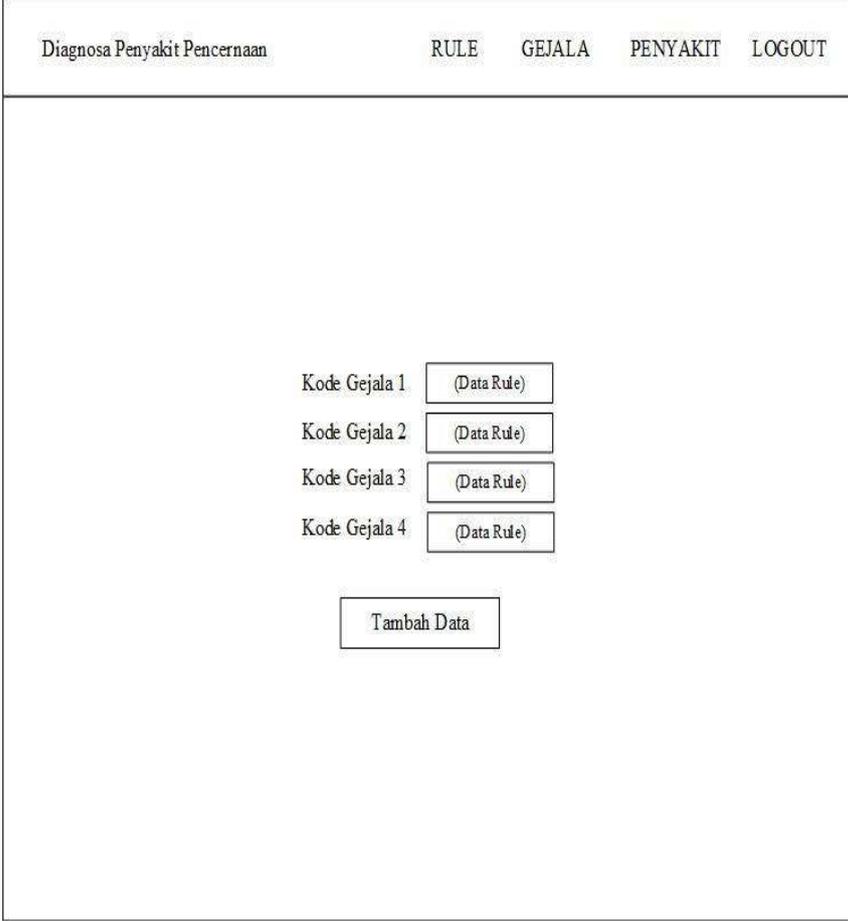
k. *Design Interface Halaman Update Penyakit*



The image shows a web interface for updating disease information. At the top, there is a navigation bar with the text "Diagnosa Penyakit Pencernaan" on the left and four menu items: "RULE", "GEJALA", "PENYAKIT", and "LOGOUT". Below the navigation bar is a large white area containing four input fields arranged horizontally. The first three fields are labeled "(Nama Penyakit)", "(Solusi)", and "(Deskripsi)" respectively. The fourth field is a button labeled "Ubah Data".

Gambar 3.22. *Design Interface Halaman Update Penyakit.*

1. *Design Interface* Halaman Tambah Basis Pengetahuan



The image shows a web interface for adding disease knowledge. At the top, there is a navigation bar with the title "Diagnosa Penyakit Pencernaan" and four menu items: "RULE", "GEJALA", "PENYAKIT", and "LOGOUT". The main content area contains four rows of input fields, each labeled "Kode Gejala" followed by a number from 1 to 4. Each input field contains the text "(Data Rule)". Below these fields is a single button labeled "Tambah Data".

Kode Gejala	(Data Rule)
Kode Gejala 1	(Data Rule)
Kode Gejala 2	(Data Rule)
Kode Gejala 3	(Data Rule)
Kode Gejala 4	(Data Rule)

Tambah Data

Gambar 3.23. Design Interface Halaman Tambah Penyakit.

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1. Construction

Pada tahap ini penelitian berfokus pada pengkodean yang menggunakan bahasa *PHP*, dan akan dilanjutkan dengan pengujian hasil menggunakan metode *blackbox testing*. *Blackbox testing* ini merupakan metode pengujian terhadap fungsionalitas atau kegunaan, apakah berjalan sesuai dengan sesestinya atau tidak.

4.1.1 Implementasi *Hardware & Software*

Dalam tahap ini dibutuhkan spesifikasi *hardware* dan *software* untuk mengoperasikan sistem pakar. Berikut merupakan penjelasan spesifikasi *hardware* dan *software* :

4.1.1.1 *Hardware*

Hardware yang digunakan sebagai berikut :

1. *Processor* Intel(R) Core(TM) i5-4310 M CPU 2.70GHz
2. *Memory* 8 GB
3. *Hard disk* 500 GB

4.1.1.2 *Software*

Software yang digunakan sebagai berikut :

1. Sistem Operasi *Windows* 10

2. *Browser (Microsoft Edge, Google Chrome)*
3. *PHP*
4. *MySQL*
5. *Bootstrap*
6. *Apache*

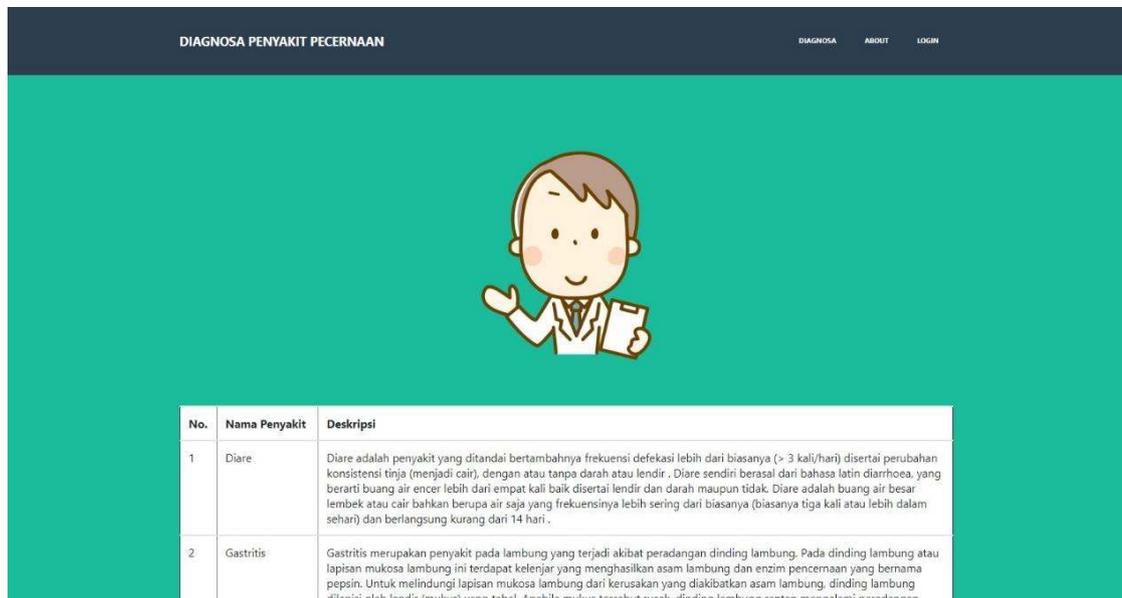
4.1.2 Implementasi Antarmuka

Pada gambar ini menjelaskan tampilan menu utama, dimana pengguna (*user*) akan memilih sesuai yang diinginkannya.



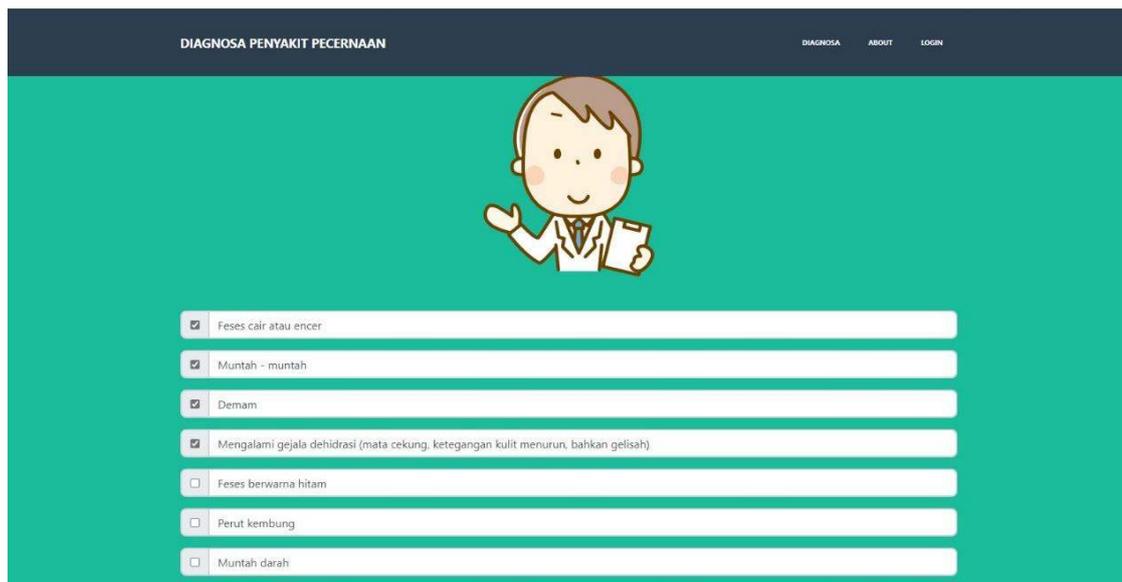
Gambar 4.1. Tampilan Menu Utama

Pada gambar ini menampilkan tampilan menu about, yang menampilkan beberapa informasi penyakit pencernaan beserta deskripsinya.



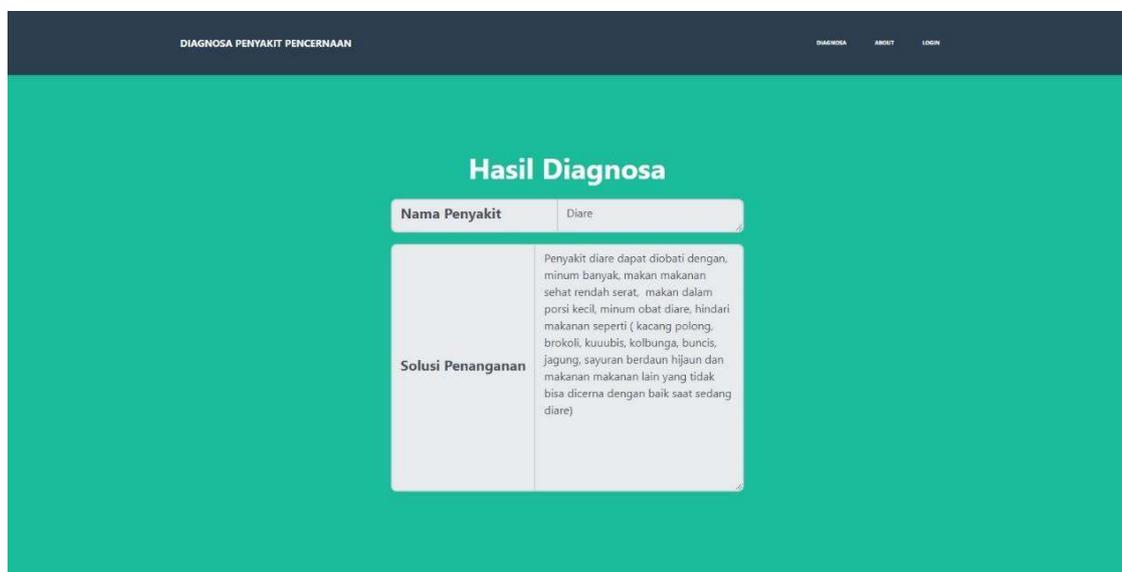
Gambar 4.2. Tampilan Menu About

Pada gambar ini menampilkan tampilan menu diagnosa, dimana pengguna (*user*) hanya bisa memilih 4 dari beberapa gejala yang di tampilkan pada menu diagnosa.



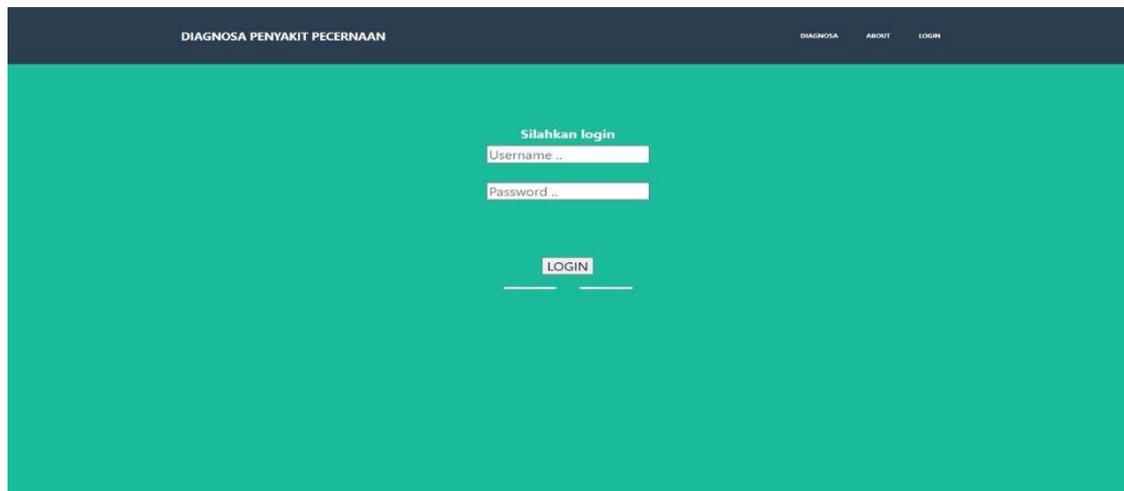
Gambar 4.3. Tampilan Menu Diagnosa

Pada gambar ini menampilkan tampilan menu hasil diagnosa menggunakan metode *Forward Chaining*, dimana pengguna (*user*) diperlihatkan hasil diagnosa berdasarkan gejala yang dipilih. Hasil Diagnosa berupa penyakit yang dialami dan solusi.



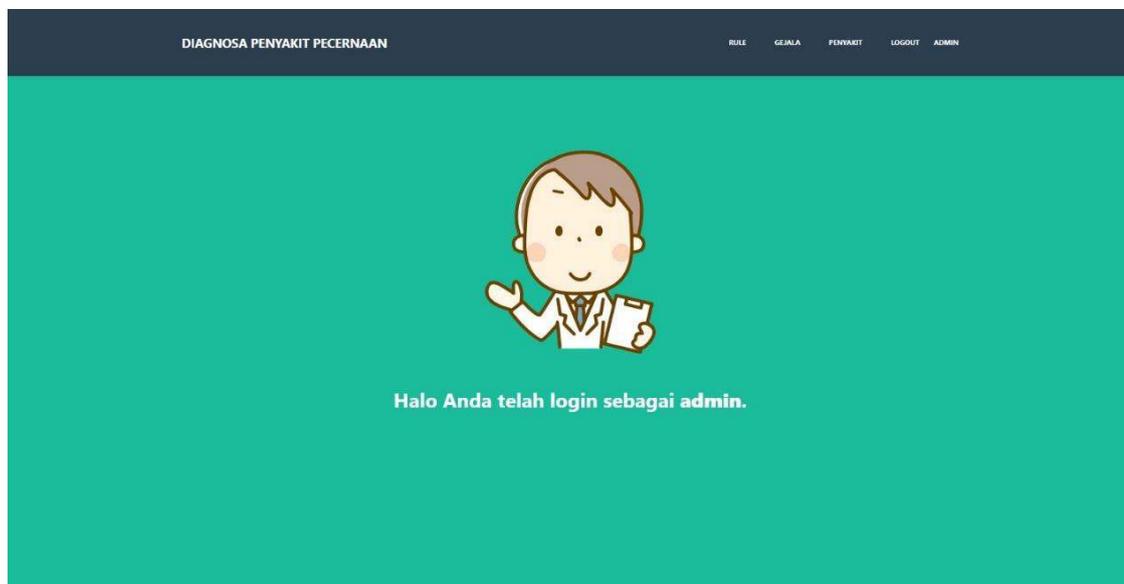
Gambar 4.4. Tampilan Menu Hasil Diagnosa

Pada gambar ini menampilkan tampilan menu login yang diperuntukan hanya untuk admin. Login dapat berhasil jika admin menginputkan username dan password dengan benar.



Gambar 4.5. Tampilan Menu Login Admin

Pada gambar ini menampilkan tampilan menu admin, tampilan menu admin akan berhasil diakses jika proses login berhasil.



Gambar 4.6. Tampilan Menu Admin

Pada gambar ini menampilkan tampilan menu *rule*, dimana admin dapat mengelola data-data basis pengetahuan.



No.	G001	G002	G003	G004	G005	G006	G007	G008	G009	G010	G011	G012	G013	G014	G015	G016	G017	G018	G019	Aksi			
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Tambah	Ubah	Hapus
2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Tambah	Ubah	Hapus
3	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	Tambah	Ubah	Hapus
4	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	Tambah	Ubah	Hapus
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	Tambah	Ubah	Hapus

Gambar 4.7. Tampilan Menu Rule

Pada gambar ini menampilkan tampilan menu gejala, dimana admin dapat mengelola data-data gejala.

DIAGNOSA PENYAKIT PENCERNAAN

RULE GEJALA PENYAKIT LOGOUT ADMIN



Kode Gejala

No.	Kode	Gejala	Aksi	
1	G001	Feses cair atau encer	Ubah	Hapus
2	G002	Muntah - muntah	Ubah	Hapus
3	G003	Demam	Ubah	Hapus
4	G004	Mengalami gejala dehidrasi (mata cekung, ketegangan kulit menurun, bahkan gelisah)	Ubah	Hapus

Gambar 4.8. Tampilan Menu Gejala

Pada gambar ini menampilkan tampilan menu penyakit, dimana admin dapat mengelola data-data penyakit.

No.	Nama Penyakit	Deskripsi	Solusi	Aksi
1	Diare	Diare adalah penyakit yang ditandai bertambahnya frekuensi defekasi lebih dari biasanya (> 3 kali/hari) disertai perubahan konsistensi tinja (menjadi cair), dengan atau tanpa darah atau lendir. Diare sendiri berasal dari bahasa latin diarrhoea, yang berarti buang air encer lebih dari empat kali baik disertai lendir dan darah maupun tidak. Diare adalah buang air besar lembek atau cair bahkan berupa air saja yang frekuensinya lebih sering dari biasanya (biasanya tiga kali atau lebih dalam sehari) dan berlangsung kurang dari 14 hari.	Penyakit diare dapat diobati dengan, minum banyak, makan makanan sehat rendah serat, makan dalam porsi kecil, minum obat diare, hindari makanan seperti (kacang polong, brokoli, kuuubis, kolbunga, buncis, jagung, sayuran berdaun hijau dan makanan makanan lain yang tidak bisa dicerna dengan baik saat sedang diare)	Ubah Hapus

Gambar 4.9. Tampilan Menu Penyakit

4.1.3 Testing

Testing atau pengujian dilakukan untuk menjamin kualitas dan juga untuk mengetahui kekurangan dari perangkat lunak yang dibangun. Salah satu metode yang digunakan dalam pengujian adalah *blackbox*.

4.1.3.1 Testing Blackbox

Testing Blackbox digunakan untuk mengetahui fungsi-fungsi khusus dari perangkat lunak yang telah dibangun.

Tabel 4.1. Tabel Rencana Pengujian

No	Kelas Uji	Butir Pengujian	Jenis Pengujian
1.	Login Admin	Verifikasi username	<i>Blackbox</i>
		Verifikasi password	
2.	Pengolahan Data Penyakit	Tambah data penyakit	<i>Blackbox</i>
		Ubah data penyakit	
		Hapus data penyakit	
3.	Pengolahan Data Gejala	Tambah data gejala	<i>Blackbox</i>
		Ubah data gejala	
		Hapus data gejala	
4.	Pengolahan Data Basis Pengetahuan	Tambah data basis pengetahuan	<i>Blackbox</i>
		Ubah data basis pengetahuan	
		Hapus data basis pengetahuan	
5.	Proses Diagnosa	Diagnosa	<i>Blackbox</i>

Setelah dilakukan pengujian dengan metode *blackbox*, maka hasil pengujian yang didapatkan adalah:

Tabel 4.2. Tabel Hasil Pengujian *Login Admin*

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
1.	Login Admin	Username dan password sesuai data yang sudah ada dalam database	Username dan Password sesuai, dan masuk ke halaman admin		Berhasil
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
1.	Login Admin	Username dan Password tetapi tidak sesuai data yang sudah ada dalam database	Login admin tidak berhasil, dan menampilkan pesan "Username dan password tidak sesuai"		Berhasil

Lanjutan Tabel 4.2. Tabel Hasil Pengujian *Login Admin*

Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
1.	Login admin	Username dan password tetapi satu dikosongkan	Login admin tidak berhasil dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.3. Tabel Hasil Pengujian Tambah Data Penyakit

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
2.	Tambah Data Penyakit	Nama penyakit, deskripsi, solusi	Data tersimpan ke database		Berhasil
2.	Tambahh Data Penyakit	Nama penyakit, deskripsi, solusi, tetapi salah satu dikosongkan	Tidak berhasil menambahkan data, dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.4. Tabel Hasil Pengujian Ubah Data Penyakit

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
2.	Ubah Data Penyakit	Nama penyakit, deskripsi, solusi yang baru	Data berhasil diubah dan tersimpan kedalam database		Berhasil
2.	Ubah Data Penyakit	Nama penyakit, deskripsi, solusi yang baru tetapi salah satu dikosongkan	Tidak berhasil mengubah data, dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.5. Tabel Hasil Pengujian Hapus Data Penyakit

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
2.	Hapus Data Penyakit	Nama penyakit, deskripsi, solusi	Menampilkan pesan ‘anda yakin akan menghapus		Berhasil

Lanjutan Tabel 4.5. Tabel Hasil Pengujian Hapus Data Penyakit

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
		yang akan dihapus	data ini?' apabila memilih 'cancel' maka data batal dihapus. Apabila memilih 'ok' maka data akan dihapus		

Tabel 4.6. Tabel Hasil Pengujian Tambah Data Gejala

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
3.	Tambah Data Gejala	Kode gejala dan gejala	Data tersimpan ke <i>database</i>		Berhasil
3.	Tambah Data Gejala	Kode gejala dan gejala tetapi salah satu dikosongkan	Tidak berhasil menambahkan data, dan menampilkan		Berhasil

Lanjutan Tabel 4.6. Tabel Hasil Pengujian Tambah Data Gejala

Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
			pesan “isi data dengan benar”		

Tabel 4.7. Tabel Hasil Pengujian Ubah Data Gejala

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
3.	Ubah Data Gejala	Kode gejala dan gejala yang baru	Data berhasil diubah dan tersimpan kedalam <i>database</i>		Berhasil
3.	Ubah Data Gejala	Kode gejala dan gejala tetapi salah satu dikosongkan	Tidak berhasil mengubah data, dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.8. Tabel Hasil Pengujian Hapus Data Gejala

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
3.	Hapus Data Gejala	Kode gejala dan gejala yang akan dihapus	Menampilkan pesan 'anda yakin akan menghapus data ini?' apabila memilih 'cancel' maka data batal dihapus. Apabila memilih 'ok' maka data akan dihapus		Berhasil

Tabel 4.9. Tabel Hasil Pengujian Tambah Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
4.	Tambah Data Basis Pengetahuan	Mengisi nilai dengan (1 atau 0)	Data berhasil ditambah dan tersimpan		Berhasil

Lanjutan Tabel 4.9. Tabel Hasil Pengujian Tambah Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
			kedalam <i>database</i>		
4.	Tambah Data Gejala	Mengisi nilai dengan (1 atau 0) tetapi salah satu kolom dikosongkan	Tidak berhasil mengubah data, dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.10. Tabel Hasil Pengujian Ubah Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
4.	Ubah Data Basis Pengetahuan	Mengisi nilai basis pengetahuan yang baru	Data berhasil diubah dan tersimpan kedalam <i>database</i>		Berhasil

Lanjutan Tabel 4.10. Tabel Hasil Pengujian Ubah Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
4.	Ubah Data Basis Pengetahuan	Mengisi nilai basis pengetahuan yang baru tetapi salah satu kolom dikosongkan	Tidak berhasil mengubah data, dan menampilkan pesan “isi data dengan benar”		Berhasil

Tabel 4.11. Tabel Hasil Pengujian Hapus Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
4.	Hapus Data Basis Pengetahuan	Basis pengetahuan yang akan dihapus	Menampilkan pesan ‘anda yakin akan menghapus data ini?’ apabila memilih		Berhasil

Lanjutan Tabel 4.11. Tabel Hasil Pengujian Hapus Data Basis Pengetahuan

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
			'cancel' maka data batal dihapus. Apabila memilih 'ok' maka data akan dihapus		

Tabel 4.12. Tabel Hasil Pengujian Diagnosa

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
5.	Diagnosa	Memilih gejala yang dialami pengguna	Data diperiksa berdasarkan tabel basis pengetahuan dan akan menampilkan hasil berupa		Berhasil

Lanjutan Tabel 4.12. Tabel Hasil Pengujian Diagnosa

Kasus dan Hasil Uji (Data Benar)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
			nama penyakit, dan solusi		
Kasus dan Hasil Uji (Data Salah)					
No.	Kelas Uji	Input	Hasil yang Diharapkan	Output	Kesimpulan
5.	Diagnosa	Memilih gejala yang dialami pengguna, tapi tidak sesuai dengan basis pengetahuan yang ada	Tidak terjadi proses diagnosa, menampilkan menu error		Berhasil

4.1.3.2 *Testing* Klasifikasi Penyakit Kejiwaan

Tahap ini merupakan tahapan pengujian seberapa besar kemampuan sistem dalam mengklasifikasi penyakit pencernaan, dimana pada penelitian ini penyakit pencernaan yang telah diklasifikasi akan muncul hasil diagnosis dan besar akurasi terhadap kesamaan penyakit pencernaan yang diuji dengan penyakit pencernaan yang ada di *database*.

Berikut merupakan kelas uji penyakit pencernaan:

Tabel 4.13. Tabel rencana *Testing* Klasifikasi Penyakit Pencernaan

No.	Kelas Uji
1.	Crohn
2.	Gastritis Akut
3.	Gastritis Kronis
4.	Maag
5.	Tukak Lambung
6.	Gastroporesis
7.	Gastroenteristis
8.	Usus Buntu
9.	Tifus
10.	Gerd
11.	Iritasi Usus Besar

4.1.4 Kesimpulan *Testing*

Berdasarkan hasil *testing* yang sudah dilakukan, dapat ditarik kesimpulan yaitu sistem pakar sudah berjalan sesuai yang diharapkan, tetapi tidak menutup kemungkinan terjadi kesalahan pada saat sistem pakar digunakan.

BAB V

PENUTUP

5.1 Kesimpulan

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

1. Dengan adanya sistem pakar diagnosis penyakit pencernaan berbasis web ini diharapkan dapat menghasilkan sebuah sistem yang dapat mendiagnosis penyakit pencernaan berdasarkan gejala-gejala yang muncul menggunakan metode *forward chaining*
2. Diharapkan dapat memberikan informasi kepada pasien, dengan cara yang mudah dan tidak terbatas oleh waktu.

5.2 Saran

Dengan adanya kesimpulan diatas, ada beberapa saran yang dapat dikemukakan sebagai bahan pertimbangan lebih lanjut guna meningkatkan produktifitas kerja dari Sistem Pakar Diagnosis Penyakit Pencernaan Menggunakan Metode Forward Chaining Berbasis Web ini.

1. Pada pengembangan selanjutnya diharapkan adanya penambahan pada hasil seperti obat yang harus dikonsumsi untuk gejala-gejala awal penyakit.

2. Diharapkan perbanyak data tentang penyakit, gejala, dan data-data yang berhubungan dengan system pakar.

DAFTAR PUSTAKA

- Ariawan, P. ., Sanjaya, D. . and Divayana, D. G. . (2016) ‘An evaluation of the implementation of practice teaching program for prospective teachers at Ganesha University of Education based on CIPP-forward chaining’, *IJARAI International Journal of Advanced Research in Artificial Intelligence*, 5(2), pp. 1–5. Available at: www.ijarai.thesai.org.
- Astuti, Y. (2016) ‘Good Governance Pengelolaan Keuangan Desa Menyongsong Berlakunya Undang-Undang No. 6 Tahun 2014.’, *Jurnal Akuntansi dan Keuangan Indonesia*, pp. 1–14.
- Buana (2014) ‘Jago pemrograman PHP’, *Dunia Komputer, Jakarta, Indonesia*.
- Fadlil, I. (2013) ‘Sistem pakar untuk mendiagnosa penyakit saluran pencernaan menggunakan metode Dempster Shafer 1’, *Jurnal Sarjana Teknik Informatika*, 1, pp. 32–41.
- Fathansyah (2015) ‘Basis Data’, *Bandung: Informatika Bandung*.
- Febriansyah (2008) ‘Buku Ajaran Intelegensi Buatan’, *Febriansyah, Palembang*.
- Huffman (1994) ‘Health Information Management’, *Berwyn Illionis:Physicians’record company*, Edisi 10.

- Irawan (2012) 'Belajar Interaktif Javascript untuk Orang Awam Edisi ke-2',
Palembang: Maxikom.
- Jogiyanto, H. M. (2005) 'Sistem Teknologi Informasi: Pendekatan Terintegrasi: Konsep Dasar, Teknologi, Aplikasi, Pengembangan dan Pengelolaan.', *ANDI OFFSET, Yogyakarta*, Edisi-2.
- Kartini (2013) 'Analisis & Desain Sistem Informasi', *Jakarta, Ilmu Komputer*, pp. 26–27.
- Komputer, W. (2010) 'Panduan Belajar MySQL Database Server', *Jakarta: Media Kita.*
- Kusumadewi, S. (2003) 'Artificial Intelligence', *Yogyakarta: Graha Ilmu.*
- Lestari (2012) 'Definisi Sistem Pakar', *Arsip Teknik Informatika.*
- Ma'rifati, - Imam Soleh and Kesuma, C. (2018) 'Pengembangan Sistem Pakar Mendeteksi Penyakit Pencernaan Menggunakan Metode Naive Bayes Berbasis Web', *Evolusi : Jurnal Sains dan Manajemen*, 6(1), pp. 41–48. doi: 10.31294/evolusi.v6i1.3543.
- Nugroho (2013) 'Dasar Pemograman Web PHP', *Yogyakarta : Gava Media.*
- Nur & Salihah, dkk (2015) 'Sistem Pakar Penentuan Bakat Anak dengan Menggunakan Metode Forward Chaining', *Jurnal Rekayasa dan Manajemen Sistem Informasi*, Vol, Nomor.

Oktavian (2013) 'Membuat website powerfull menggunakan PHP', *Yogyakarta: MEDIA KOM.*

Prasetio (2012) 'Buku Pintar Pemrograman Web', *Jakarta: Mediakita.*

Pressman (2015) 'Rekayasa Perangkat Lunak', *Pendekatan Praktisi Buku I. Yogyakarta: Andi.*

Priyadi (2014) 'Kolaborasi SQL dan ERD dalam Implementasi Database', *Yogyakarta : ANDI.*

Rafi'i (2008) 'Cara Cepat Membuat Website', *Surabaya: Tiara Aksa.*

Rahman (2013) 'Cara Gampang Bikin CMS PHP Tanpa Ngoding', *Jakarta: PT. Transmedia.*

Ramli & Pamoentjak (1990) 'Kamus Kedokteran', *Jakarta: Djambatan.*

Saputra (2012) 'Web Tips PHP, HTML5 dan CSS3', *Jasakom, Jakarta.*

Sibero (2013) 'Web programming power pack', *MediaKom, Yogyakarta.*

Simarmata, 2007 (2007) 'Perancangan Basis data', *Andi. Yogyakarta.*

Situmorang, T. . *et al.* (2016) 'Faktor - Faktor Yang Berhubungan Dengan Kejadian Rsu Anutapura Palu', *Jurnal Kesehatan Tadulako*, 2(1), pp. 34–44. Available at: <http://jurnal.fk.untad.ac.id/index.php/htj/article/view/21>.

Solichin (2016) 'Pemrograman Web dengan PHP dan MySQL'. Edisi-1. *Jakarta:*

Budi Luhur.

Sukamto & Shalahuddin (2013) 'Analisa dan Desain Sistem Informasi',
Yogyakarta: Andi Offset.

Sukamto & Shalahuddin (2014) 'Rekayasa Perangkat Lunak Terstruktur Dan
Berorientasi Objek', *Bandung: Informatika Bandung.*

Supono & Putratama (2016) 'Pemograman Web Dengan Menggunakan PHP dan
Framework Codeigniter', *Yogyakarta: Deepublish (Grup Penerbitan CV
Budi Utama).*

Tambayong (2001) 'Anatomi & Fisiologi untuk Keperawatan', *Buku Kedokteran
IGC.*

LAMPIRAN

Konsultasi.php

```
<?php
include "librari/inc.koneksidb.php";

$RbPilih  = $_REQUEST['RbPilih'];
$TxtKdGejala  = $_REQUEST['TxtKdGejala'];

function AddTmpAnalisa($kdgejala, $NOIP) {
    global $koneksi;
    $sql_sakit = "SELECT relasi.* FROM relasi,tmp_penyakit
        WHERE relasi.kd_penyakit=tmp_penyakit.kd_penyakit
        AND noip='$NOIP' ORDER BY relasi.kd_penyakit,relasi.kd_gejala";
    $qry_sakit = mysqli_query($koneksi, $sql_sakit);
    while ($data_sakit = mysqli_fetch_array($qry_sakit)) {
        $sqltmp = "INSERT INTO tmp_analisa (noip, kd_penyakit,kd_gejala)
            VALUES
            ('$NOIP','$data_sakit[kd_penyakit]','$data_sakit[kd_gejala]')";
        mysqli_query($koneksi, $sqltmp);
    }
}

function AddTmpGejala($kdgejala, $NOIP) {
    global $koneksi;
    $sql_gejala = "INSERT INTO tmp_gejala (noip,kd_gejala) VALUES
    ('$NOIP','$kdgejala')";
    mysqli_query($koneksi, $sql_gejala);
}

function DelTmpSakit($NOIP) {
    global $koneksi;
    $sql_del = "DELETE FROM tmp_penyakit WHERE noip='$NOIP'";
    mysqli_query($koneksi, $sql_del);
}
```

```

function DelTmpAnlisa($NOIP) {
    global $koneksi;
    $sql_del = "DELETE FROM tmp_analisa WHERE noip='$NOIP'";
    mysqli_query($koneksi, $sql_del);
}

if ($RbPilih == "YA") {
    $sql_analisa = "SELECT * FROM tmp_analisa where noip='$NOIP' ";
    $qry_analisa = mysqli_query($koneksi, $sql_analisa);
    $data_cek = mysqli_num_rows($qry_analisa);
    if ($data_cek >= 1) {

        DelTmpSakit($NOIP);
        $sql_tmp = "SELECT * FROM tmp_analisa
            WHERE kd_gejala='$TxtKdGejala'
            AND noip='$NOIP'";
        $qry_tmp = mysqli_query($koneksi, $sql_tmp);
        while ($data_tmp = mysqli_fetch_array($qry_tmp)) {
            $sql_rsakit = "SELECT * FROM relasi
                WHERE kd_penyakit='$data_tmp[kd_penyakit]'
                GROUP BY kd_penyakit";
            $qry_rsakit = mysqli_query($koneksi, $sql_rsakit);
            while ($data_rsakit = mysqli_fetch_array($qry_rsakit)) {

                $sql_input = "INSERT INTO tmp_penyakit (noip,kd_penyakit)
                    VALUES ('$NOIP','$data_rsakit[kd_penyakit]')";
                mysqli_query($koneksi, $sql_input);
            }
        }

        DelTmpAnlisa($NOIP);
        AddTmpAnalisa($TxtKdGejala, $NOIP);
        AddTmpGejala($TxtKdGejala, $NOIP);
    }
    else {

        $sql_rgejala = "SELECT * FROM relasi WHERE
            kd_gejala='$TxtKdGejala'";
        $qry_rgejala = mysqli_query($koneksi, $sql_rgejala);
        while ($data_rgejala = mysqli_fetch_array($qry_rgejala)) {
            $sql_rsakit = "SELECT * FROM relasi
                WHERE kd_penyakit='$data_rgejala[kd_penyakit]'
                GROUP BY kd_penyakit";
        }
    }
}

```

```

    $qry_rsakit = mysqli_query($koneksi, $sql_rsakit);
    while ($data_rsakit = mysqli_fetch_array($qry_rsakit)) {

        $sql_input = "INSERT INTO tmp_penyakit (noip,kd_penyakit)
            VALUES ('$NOIP','$data_rsakit[kd_penyakit]')";
        mysqli_query($koneksi, $sql_input);
    }

    AddTmpAnalisa($TxtKdGejala, $NOIP);
    AddTmpGejala($TxtKdGejala, $NOIP);
}
echo "<meta http-equiv='refresh' content='0; url=index.php?page=konsul'>";
}

if ($RbPilih == "TIDAK") {
    $sql_analisa = "SELECT * FROM tmp_analisa where noip='$NOIP' ";
    $qry_analisa = mysqli_query($koneksi, $sql_analisa);
    $data_cek = mysqli_num_rows($qry_analisa);
    if ($data_cek >= 1) {

        $sql_relasi = "SELECT * FROM tmp_analisa WHERE
kd_gejala='$TxtKdGejala'";
        $qry_relasi = mysqli_query($koneksi, $sql_relasi);
        while($hsl_relasi = mysqli_fetch_array($qry_relasi)){

            $sql_deltmp = "DELETE FROM tmp_analisa
                WHERE kd_penyakit='$hsl_relasi[kd_penyakit]'
                AND noip='$NOIP'";
            mysqli_query($koneksi, $sql_deltmp);

            $sql_deltmp2 = "DELETE FROM tmp_penyakit
                WHERE kd_penyakit='$hsl_relasi[kd_penyakit]'
                AND noip='$NOIP'";
            mysqli_query($koneksi, $sql_deltmp2);
        }
    }
} else {

    $sql_relasi= "SELECT * FROM relasi ORDER BY kd_penyakit,kd_gejala";
    $qry_relasi= mysqli_query($koneksi, $sql_relasi);
    while($hsl_relasi=mysqli_fetch_array($qry_relasi)){

```

```

    $sql_intmp = "INSERT INTO tmp_analisa (noip, kd_penyakit,kd_gejala)
        VALUES ('$NOIP','$hsl_relasi[kd_penyakit]',
            '$hsl_relasi[kd_gejala]')";
    mysqli_query($koneksi, $sql_intmp);

    $sql_intmp2 = "INSERT INTO tmp_penyakit(noip,kd_penyakit)
        VALUES ('$NOIP','$hsl_relasi[kd_penyakit]')";
    mysqli_query($koneksi, $sql_intmp2);
}

    $sql_relasi2 = "SELECT * FROM relasi WHERE
kd_gejala='$TxtKdGejala'";
    $qry_relasi2 = mysqli_query($koneksi, $sql_relasi2);
    while($hsl_relasi2 = mysqli_fetch_array($qry_relasi2)){
        $sql_deltmp = "DELETE FROM tmp_analisa
            WHERE kd_penyakit='$hsl_relasi2[kd_penyakit]'
            AND noip='$NOIP'";
        mysqli_query($koneksi, $sql_deltmp);

        $sql_deltmp2 = "DELETE FROM tmp_penyakit
            WHERE kd_penyakit='$hsl_relasi2[kd_penyakit]'
            AND noip='$NOIP'";
        mysqli_query($koneksi, $sql_deltmp2);
    }

}
echo "<meta http-equiv='refresh' content='0'; url=index.php?page=konsul'>";
}

    $sql_cekh = "SELECT * FROM tmp_analisa
        WHERE noip='$NOIP'
        GROUP BY kd_penyakit";
    $qry_cekh = mysqli_query($koneksi, $sql_cekh);
    $hsl_cekh = mysqli_num_rows($qry_cekh);

    if ($hsl_cekh == 0){

        ?>
        <script>
            confirmed = window.confirm('Anda Baik-Baik Saja')

```

```

        if (confirmed) {
            window.location = "index.php?page=daftar";
        }

</script>
<?php
}
?>

AnalisisHasil.php

<?php
include "librari/inc.koneksidb.php";

$NOIP = $_SERVER['REMOTE_ADDR'];
$sql = "SELECT analisa_hasil.*, penyakit.*
FROM analisa_hasil, penyakit
WHERE penyakit.kd_penyakit=analisa_hasil.kd_penyakit
AND analisa_hasil.noip='$NOIP'
ORDER BY analisa_hasil.id DESC LIMIT 1";
$qry = mysqli_query($koneksi, $sql)
    or die ("Query Hasil salam".mysqli_error());
$data= mysqli_fetch_array($qry);
if ($data['kelamin']=="P") {
    $kelamin = "Pria";
}
else {
    $kelamin = "Wanita";
}
?>
<html>
<head>
<title>Hasil Analisa Pasien</title>
</head>
<body>
<table width="100%" border="0" cellpadding="2" cellspacing="1"
bgcolor="#DBEAF5">
    <tr align="center">
        <td colspan="2" bgcolor="#22B5DD"><b>HASIL ANALISA PENYAKIT
KULIT</b></td>
    </tr>
    <tr>
        <td colspan="2"><b>DATA PASIEN :</b></td>
    </tr>

```

```

<tr bgcolor="#FFFFFF">
  <td width="86">Nama</td>
  <td width="689"><?php echo $data['nama']; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td>Kelamin</td>
  <td><?php echo $kelamin; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td>Alamat</td>
  <td><?php echo $data['alamat']; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td>Pekerjaan</td>
  <td><?php echo $data['pekerjaan']; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td colspan="2"><b>HASIL ANALISA TERAKHIR :</b></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td>Penyakit</td>
  <td><?php echo $data['nm_penyakit']; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td valign="top">Penyebab</td>
  <td><?php echo $data['penyebab']; ?></td>
</tr>
<tr>
<tr bgcolor="#FFFFFF">
  <td valign="top">Gejala</td>
  <td>
    <?php
    $sql_gejala = "SELECT gejala.* FROM gejala, relasi
      WHERE gejala.kd_gejala=relasi.kd_gejala
      AND relasi.kd_penyakit='$data[kd_penyakit]'";
    $qry_gejala = mysqli_query($koneksi, $sql_gejala);
    while ($hsl_gejala=mysqli_fetch_array($qry_gejala)) {
    $i++;
      echo "$i . $hsl_gejala[nm_gejala] <br>";
    }
  </td>
</tr>

```

```

?>
</td>
</tr>
<tr bgcolor="#FFFFFF">
  <td valign="top">Keterangan</td>
  <td><?php echo $data['keterangan']; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
  <td valign="top">Solusi</td>
  <td><?php echo $data['solusi']; ?></td>
</tr>
<tr>
  <td>
    <script>
      function disp_confirm(){
        var r=confirm("Apakah Anda Akan Mengulangi Konsultasi...!!!")
        if (r==true){
          window.location = "index.php?page=back"
        }
      }
    </script>
    <input type="button" onclick="disp_confirm()" value="kembali">

    <script>
      function keluar(){
        var k=confirm("Apakah Anda Akan Logout...!!!")
        if (k==true){
          window.location = "index.php?page=daftar"
        }
      }
    </script>
    <input type="button" onclick="keluar()" value="Logout">

  </td>
</tr>
</table>
</body>
</html>

```