

**PENERAPAN SISTEM PAKAR BERBASIS WEB DENGAN  
METODE *FORWARD CHAINING* UNTUK  
MENGIDENTIFIKASI PENYAKIT PADA AYAM**

**Skripsi**

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi  
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)  
dalam Teknologi Informasi



Diajukan Oleh :

**MAHENDRA ADRIANOVA RUSTIYONO**

NIM : 1908096038

**PROGRAM STUDI S-1 TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG  
2022**

## PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini :

Nama : Mahendra Adrianova Rustiyono

NIM : 1908096038

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

Penerapan Sistem Pakar Berbasis Web dengan Metode *Forward Chaining* untuk Mengidentifikasi Penyakit pada Ayam

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 24 Juni 2023

Pembuat Pernyataan,



Mahendra Adrianova Rustiyono

NIM : 1908096038



KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI WALISONGO  
FAKULTAS SAINS DAN TEKNOLOGI  
Jl. Prof. Dr. Hamka Ngaliyan Semarang  
Telp.024-7601295 Fax.7615387

---

**PENGESAHAN**

Naskah proposal skripsi berikut ini:

Judul : Penerapan Sistem Pakar Berbasis Web dengan  
Metode *Forward Chaining* untuk Mengidentifikasi  
Penyakit pada Ayam

Penulis : **Mahendra Adrianova Rustiyono**

NIM : 1908096038

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang komperehensif oleh Dewan  
Penguji Fakultas Sains dan Teknologi UIN Walisongo.

Semarang, 27 September 2023

DEWAN PENGUJI

**Penguji I**

Nur Cahyo H.W., S.T., M.Kom  
NIP. 197312222006041001

**Penguji III**

Siti Nur'aini, M.Kom  
NIP. 198401312018012001

**Pembimbing I**

Dr. Masy Ari Ulinuha, M.T.  
NIP. 198108122011011007

**Penguji II**

Mokhamad Ikhlil Mustofa, M.Kom  
NIP.198808072019031010

**Penguji IV**

Hery Mustofa, M.Kom  
NIP.198703172019031007

**Pembimbing II**

Mokhamad Ikhlil Mustofa, M.Kom  
NIP.198808072019031010

## NOTA PEMBIMBING

Semarang, 24 Juni 2023

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

Assalamu'alaikum.wr.wb

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Penerapan Sistem Pakar Berbasis Web dengan  
Metode *Forward Chaining* untuk Mengidentifikasi  
Penyakit pada Ayam  
Penulis : Mahendra Adrianova Rustiyono  
NIM : 1908096038  
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum.wr.wb

Pembimbing I,



Dr. Masy Ari Ulinuha, M.T.

NIP. 198108122011011007

## NOTA PEMBIMBING II

Semarang, 24 Juni 2023

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

Assalamu'alaikum.wr.wb

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Penerapan Sistem Pakar Berbasis Web dengan Metode *Forward Chaining* untuk Mengidentifikasi Penyakit pada Ayam

Penulis : Mahendra Adrianova Rustiyono

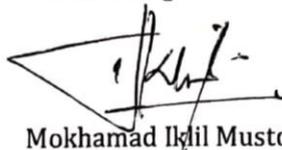
NIM : 1908096038

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum.wr.wb

Pembimbing II,



Mokhammad Ilkil Mustofa, M.Kom

NIP.198808072019031010

## KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya atas selesainya pembuatan laporan skripsi dengan judul “Penerapan Sistem Pakar Berbasis Web dengan Metode *Forward Chaining* untuk Mengidentifikasi Penyakit pada Ayam”. Laporan ini disusun sebagai salah satu syarat menyelesaikan program pendidikan Strata-1 Program Studi Teknologi Informasi di Universitas Islam Negeri Walisongo Semarang.

Penyusunan laporan ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis hendak mengucapkan terima kasih kepada :

1. Kedua orang tua yang selalu mendukung dan mendoakan penulis dalam menyelesaikan skripsi,
2. Teman-teman penulis yang selalu mendukung penulis dalam menyelesaikan skripsi yang tidak bisa penulis sebutkan satu-satu,
3. Bapak dan Ibu dosen UIN Walisongo Semarang.
4. Bapak Nur Cahyo Hendro Wibowo S.T., M.Kom. Selaku Ketua Program Studi Teknologi Informasi,
5. Bapak Dr. Masy Ari Ulinuha, M.T. selaku Pembimbing I yang selalu membantu penulis,
6. Bapak Mokhamad Iklil Mustofa, M.Kom selaku Pembimbing II yang juga selalu membantu penulis,

7. Andri Setiawan S.Pt yang selalu membantu penulis,

8. Seluruh pihak yang tidak bisa penulis sebutkan satu persatu.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang tidak dapat disebutkan satu persatu dan terlibat dalam penyusunan laporan skripsi sehingga dapat diselesaikan dengan baik. Semoga penyusunan skripsi ini dapat memberikan manfaat kepada semua pihak, terutama di bidang pendidikan.

Semarang, 24 Juni 2023

Penulis,



Mahendra Arianova Rustiyono

NIM: 1908096038

## ABSTRAK

Komoditas ayam di Indonesia mempunyai prospek pasar yang sangat baik karena didukung oleh karakteristik produk yang dapat diterima oleh semua lapisan masyarakat, karena harga daging ayam yang relatif lebih murah (dibandingkan dengan daging sapi maupun kambing) dengan akses yang mudah diperoleh karena sudah menyebar di seluruh wilayah tanah air.

Berternak ayam menjadi salah satu pilihan dalam memulai usaha, terutama berternak ayam ras petelur karena sangat potensial dikembangkan untuk diambil telur dan dagingnya dengan produksi utama adalah telur. Akan tetapi tentu akan ada banyak masalah yang harus dihadapi terutama masalah penyakit. Maka dari itu dibutuhkan sebuah alternatif yang dapat membantu dalam mengetahui berbagai penyakit ayam, yaitu sistem pakar berbasis web dengan metode *Forward Chaining*, yaitu metode yang bekerja dengan mengumpulkan fakta-fakta yang ada menuju kepada

Sistem pakar ini nantinya akan dibuat dengan menggunakan bahasa pemrograman PHP dan juga MySQL sebagai databasenya. Sehingga aplikasi sistem pakar nanti akan berbentuk sebuah website.

**Kata Kunci :** Ternak Ayam, Sistem Pakar, *Forward Chaining*

## DAFTAR ISI

PERNYATAAN KEASLIAN .....	ii
PENGESAHAN.....	iii
NOTA PEMBIMBING I.....	iv
NOTA PEMBIMBING II.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah.....	3
C. Batasan Masalah.....	3
D. Tujuan Penelitian.....	3
E. Manfaat Penelitian.....	4
<b>BAB II LANDASAN PUSTAKA .....</b>	<b>5</b>
A. Kajian Pustaka .....	5
1. PHP.....	5
2. Forward Chaining.....	5
3. Waterfall.....	6
4. XAMPP.....	7
5. MySQL.....	7
6. Penyakit Ayam .....	8
B. Kajian Penelitian yang Relevan.....	11
<b>BAB III METODE PENELITIAN .....</b>	<b>15</b>
A. Metode Pengumpulan data .....	15
1. Wawancara .....	15
2. Studi Pustaka .....	15
B. Metode Pengembangan Sistem.....	15
1. Waterfall.....	15
A. Requirement Analysis.....	16
B. System Design.....	17
C. Implementation.....	19
D. Integration and Testing.....	19
E. Operation and Maintenance .....	19

<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN .....</b>	<b>21</b>
A. Requirement Analysis.....	21
F. Kebutuhan Sistem .....	21
G. Tabel Basis Pengetahuan.....	23
B. System Design.....	31
1. Data Flow Diagram .....	32
2. Wireframe.....	33
3. Kerangka Sistem Pakar.....	39
4. Flowchart.....	41
C. Implementation .....	42
a. Proses Coding.....	42
b. Tampilan Hasil Coding.....	62
D. Integration and Testing .....	68
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>85</b>
A. Kesimpulan .....	85
B. Saran .....	85
DAFTAR PUSTAKA.....	86

## DAFTAR GAMBAR

Gambar 1.0 Tahapan Metode Waterfall.....	16
Gambar 2.1 Diagram Konteks .....	32
Gambar 2.2 Data Flow Diagram Level 0 .....	32
Gambar 3.1 Wireframe Halaman Awal Website.....	33
Gambar 3.2 Wireframe Halaman Daftar Akun .....	33
Gambar 3.3 Wireframe Halaman Login .....	34
Gambar 3.4 Wireframe Halaman Admin.....	34
Gambar 3.5 Wireframe Halaman User .....	35
Gambar 3.6 Wireframe Halaman Info Akun .....	35
Gambar 3.7 Wireframe Halaman Tambah Data Penyakit .....	36
Gambar 3.8 Wireframe Halaman Cek Penyakit .....	36
Gambar 3.9 Wireframe Halaman Daftar Penyakit.....	37
Gambar 3.10 Wireframe Halaman Edit Data Penyakit .....	37
Gambar 3.11 Wireframe Halaman Hasil Diagnosa.....	38
Gambar 3.12 Wireframe Halaman Hasil Error.....	38
Gambar 4.0 Kerangka Sistem Pakar .....	39
Gambar 5.0 Flowchart.....	41
Gambar 6.1 Coding Proses Login (1).....	42
Gambar 6.2 Coding Proses Login (2).....	44
Gambar 6.3 Coding Proses Registrasi.....	44
Gambar 6.4 Coding Tombol submit-mnk.....	45
Gambar 6.5 Coding Tambah Data Penyakit (1) .....	46
Gambar 6.6 Coding Tambah Data Penyakit (2) .....	46
Gambar 6.7 Coding Tambah Data Penyakit (3) .....	47
Gambar 6.8 Coding URL dengan parameter del .....	48
Gambar 6.9 Coding Hapus Data Penyakit .....	48
Gambar 6.10 Coding Edit Data Penyakit (1) .....	49
Gambar 6.11 Coding Edit Data Penyakit (2) .....	50
Gambar 6.12 Coding Tombol edit-gjl.....	50
Gambar 6.13 Coding Edit Data Penyakit (3) .....	50
Gambar 6.14 Coding koneksi.php.....	51
Gambar 6.15 Coding Cek Penyakit (1) .....	52
Gambar 6.16 Coding Cek Penyakit (2) .....	53
Gambar 6.17 Coding Cek Penyakit (3) .....	54
Gambar 6.18 Coding Cek Penyakit (4) .....	55

Gambar 6.19 Coding Cek Penyakit (5) .....	56
Gambar 6.20 Coding Ubah Password .....	57
Gambar 6.21 Coding Tombol edit-user .....	58
Gambar 6.22 Coding Hapus Akun User .....	59
Gambar 7.1 Structure Database db_sp .....	60
Gambar 7.2 Tabel anggota .....	60
Gambar 7.3 Tabel tb_gejala .....	61
Gambar 7.4 Tabel tb_penyakit .....	61
Gambar 7.5 Tabel tb_rule .....	62
Gambar 8.1 Tampilan Halaman Awal .....	62
Gambar 8.2 Tampilan Halaman Login .....	63
Gambar 8.3 Tampilan Halaman Buat Akun .....	63
Gambar 8.4 Tampilan Halaman User .....	64
Gambar 8.5 Tampilan Halaman Cek Penyakit .....	64
Gambar 8.6 Tampilan Halaman Hasil Diagnosa .....	65
Gambar 8.7 Tampilan Halaman Hasil Error .....	65
Gambar 8.8 Tampilan Halaman Info Akun User .....	65
Gambar 8.9 Tampilan Halaman Admin .....	66
Gambar 8.10 Tampilan Halaman Daftar Penyakit .....	66
Gambar 8.11 Tampilan Halaman Tambah Data Penyakit .....	67
Gambar 8.12 Tampilan Halaman Edit Data Penyakit .....	67
Gambar 9.1 Uji Pilih Gejala (1) .....	82
Gambar 9.2 Uji Pilih Gejala (2) .....	83
Gambar 9.3 Hasil Pengujian .....	83

## DAFTAR TABEL

Tabel 1.0 Tabel Kebutuhan Sistem.....	21
Tabel 2.0 Tabel Data Penyakit Hasil Wawancara .....	22
Tabel 3.1 Tabel Penyakit .....	23
Tabel 3.2 Tabel Gejala .....	25
Tabel 3.3 Tabel Relasi Penyakit dan Gejala .....	27
Tabel 4.1 Tabel Pengujian Registrasi Akun .....	68
Tabel 4.2 Tabel Pengujian Login User .....	69
Tabel 4.3 Tabel Pengujian Login Admin.....	70
Tabel 4.4 Tabel Pengujian Cek Penyakit .....	71
Tabel 4.5 Tabel Pengujian Tambah Data Penyakit .....	73
Tabel 4.6 Tabel Pengujian Edit Data Penyakit .....	77
Tabel 4.7 Tabel Pengujian Hapus Data Penyakit .....	81
Tabel 4.8 Tabel Pengujian Ubah Password.....	82
Tabel 4.9 Tabel Pengujian Hapus Akun User .....	82
Tabel 4.10 Tabel Pengujian Logout.....	83

# BAB I

## PENDAHULUAN

### A. Latar Belakang Masalah

Pandemi Covid-19 beberapa waktu yang lalu memang menimbulkan banyak sekali dampak negatif di beberapa sektor, salah satu contohnya pada sektor perekonomian. Banyak perusahaan tidak sanggup meneruskan produktivitas usaha hingga harus lakukan Pemutusan Hubungan Kerja (PHK).

Meskipun sekarang sudah dalam masa peralihan menuju Endemi, namun masih ada sebagian pekerja yang di PHK dan dirumahkan tidak lagi dibutuhkan, sehingga ada yang memilih untuk mencoba usaha sendiri, seperti contohnya Pak Diar, di dusun Randusari yang memilih usaha berternak ayam sendiri.

Pemilihan usaha berternak ayam ini didasari oleh target pasarnya yang luas dan masa panen yang terbilang cepat, terutama berternak ayam ras petelur karena sangat potensial dikembangkan untuk diambil telur dan dagingnya dengan produksi utama adalah telur (Dewanti & Sudyono, 2017). Akan tetapi, dalam memulai usaha berternak ayam tentu tidak semudah yang dipikirkan, banyak rintangan yang perlu dihadapi terutama dalam hal penyakit ayam. Ketika ternak ayam Pak Diar terserang penyakit dan tidak tahu apa yang harus dilakukan,

menyebabkan banyak ayam yang mati, serta membuat usaha ternak ayam Pak Diar hampir mengalami kebangkrutan karena kurangnya pengetahuan dalam hal penyakit dan perawatan ayam. Setelah diperiksakan ke dokter hewan untuk mengetahui penyakit apa yang diderita ayam tersebut, Pak Diar mulai untuk belajar cara perawatan ayam dan apa saja penyebab penyakit pada ayam sehingga kedepannya dapat mengantisipasi apabila ternaknya terserang penyakit lagi. Didalam QS. at-taghabun: 11 yang berbunyi :

مَا أَصَابَ مِنْ مُصِيبَةٍ إِلَّا بِإِذْنِ اللَّهِ ۗ وَمَنْ يُؤْمِنْ بِاللَّهِ  
يَهْدِ اللَّهُ قَلْبَهُ ۚ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ

Artinya :

Tidak ada suatu musibah pun yang menimpa seseorang kecuali dengan ijin Allah; dan barangsiapa yang beriman kepada Allah niscaya Dia akan memberi petunjuk kepada hatinya. Dan Allah Maha Mengetahui segala sesuatu (QS. At-taghabun: 11).

Maka dari itu, untuk membantu mengetahui penyebab penyakit pada ayam, dibutuhkan sebuah aplikasi sistem pakar dengan tujuan untuk membantu pelaku usaha ternak ayam pemula dalam mengidentifikasi penyakit ayam, penyebabnya, cara pencegahan, penanganan serta perawatannya untuk meminimalisir

kerugian yang ditimbulkan apabila ternak ayam terserang penyakit sehingga mengakibatkan gagal panen, karena sistem pakar menyimpan pengetahuan dan keahlian para pakar sehingga memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.

## **B. Rumusan Masalah**

Berdasarkan latar belakang yang sudah dijelaskan, maka diperoleh rumusan masalah yang dapat diangkat pada penelitian ini, yaitu bagaimana cara membuat suatu sistem pakar yang dapat mengidentifikasi penyakit ayam serta penanganan dan perawatannya dengan menggunakan metode *forward chaining*.

## **C. Batasan Masalah**

Berdasarkan rumusan masalah di atas, maka dapat diperoleh batasan masalah yang dibahas pada penelitian ini antara lain sebagai berikut:

1. Penelitian ini akan menggunakan metode *forward chaining* sebagai metode pengambilan keputusan.
2. Penelitian ini hanya dibatasi pada 8 penyakit ayam yang sering terjadi pada peternakan.
3. Penelitian ini hanya dibatasi pada penyakit ayam tanpa memperhitungkan laba dan rugi usaha.

## **D. Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk menerapkan metode *forward chaining* sebagai metode pengambilan

keputusan dalam sistem pakar untuk membantu identifikasi penyakit pada ayam.

## **E. Manfaat Penelitian**

### 1. Manfaat akademis

Penelitian ini diharapkan bisa menjadi media referensi bagi penelitian selanjutnya yang menggunakan konsep dan dasar penelitian yang sama tentang sistem pakar menggunakan metode *forward chaining*.

### 2. Manfaat praktis

Penelitian ini diharapkan dapat membantu masyarakat yang ingin memulai usaha peternakan ayam namun tidak memiliki pengetahuan yang cukup tentang macam-macam penyakit ayam dan bagaimana solusi menangani penyakit tersebut.

## BAB II

### LANDASAN PUSTAKA

#### A. Kajian pustaka

##### 1. PHP

PHP (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman web berupa script yang dapat diintegrasikan dengan HTML (Anhar, 2010). PHP juga merupakan bahasa pemrograman yang didesain khusus untuk web development atau pengembangan web. Sebagai sebuah scripting language, PHP menjalankan instruksi pemrograman pada saat proses runtime. Hasil dari instruksi yang dijalankan akan berbeda tergantung data yang diproses. Script dari PHP kemudian akan diproses didalam server. Jenis server yang sering digunakan bersama dengan PHP antara lain *Apache*, *Nginx*, dan *LiteSpeed*.

##### 2. *Forward Chaining*

Metode *Forward Chaining* merupakan salah satu dari beberapa metode pengambilan keputusan yang digunakan dalam sistem pakar yang memiliki cara kerja dengan menggunakan pelacakan ke depan, yaitu teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule

tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database (Kusbianto et al., 2017), sehingga metode ini juga sering disebut “data driven”. Metode *Forward Chaining* memiliki kelebihan yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/ menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.

Berbagai struktur kaidah if-then yang menghubungkan objek atau atribut sebagai berikut: (1) IF premis THEN konklusi (2) IF masukan THEN keluaran (3) IF kondisi THEN tindakan (4) IF anteseden THEN konsekuen (5) IF data THEN hasil (6) IF tindakan THEN tujuan (7) IF aksi THEN reaksi IF sebab THEN akibat (8) IF gejala THEN diagnosa.

### 3. *Waterfall*

Model SDLC air terjun (*Waterfall*) sering juga disebut model sekuensial linier (Sequential Linear) atau alur hidup klasik (*classic life cycle*) (A.S & Salahudin, 2016). Metode pengembangan aplikasi ini menekankan pada fase yang berurutan dan sistematis. Untuk model pengembangannya sendiri, dapat dianalogikan seperti air terjun, dimana setiap tahap

harus dikerjakan secara berurutan mulai dari atas hingga ke bawah.

Penggunaan metode pengembangan waterfall pertama kali diperkenalkan oleh Herbert D. Benington di *Symposium on Advanced Programming Method for Digital Computers* pada tanggal 29 Juni 1956. Presentasi tersebut menjelaskan tentang pengembangan perangkat lunak untuk SAGE (*Semi Automatic Ground Environment*).

#### 4. XAMPP

XAMPP merupakan kepanjangan dari X (*cross platform*), A (Apache), M (MySQL/MariaDB), P (PHP), dan P (Perl), XAMPP adalah software atau aplikasi komputer yang banyak digunakan dalam dunia web developer yang juga bisa dipelajari untuk membuat website. XAMPP dikenal sebagai perangkat lunak berbasis web server yang bersifat open source, XAMPP juga mendukung berbagai sistem operasi seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. XAMPP juga dapat menggantikan peran web hosting dengan cara menyimpan file website ke dalam hosting lokal agar bisa dipanggil lewat browser.

#### 5. MySQL

MySQL adalah adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas

sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Suatu *Database Management System* (DBMS) dengan perintah dasar *Structured Query Language* (SQL) yang sering dimanfaatkan dalam pembuatan aplikasi dan website. MySQL adalah open source DBMS yang terbagi menjadi dua bentuk lisensi, yakni Free Software dan Shareware. Free Software artinya siapa saja bisa mengakses perangkat lunak tersebut, sementara itu Shareware merupakan perangkat lunak berpemilik yang mempunyai batasan atau limitasi dalam pemakaiannya. MySQL juga termasuk ke dalam suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. Metode relational database pada MySQL juga dipakai sebagai penghubung antara software dan server database.

## 6. Penyakit Ayam

### a) *Egg Drop Syndrome 76/EDS 76*

*Egg Drop Syndrome 76* merupakan penyakit infeksius organ reproduksi pada ayam di masa bertelur yang ditandai dengan penurunan produksi telur hingga mencapai

10-40%, dan gangguan pigmentasi kerabang telur.

b) *Infectious Coryza* (SNOT)

*Infectious Coryza* (SNOT) merupakan salah satu penyakit bakterial yang menyerang sistem pernafasan ayam, terutama saluran pernafasan bagian atas. Korisa menyebabkan produksi telur turun hingga 10-40%, peningkatan ayam afkir, morbiditas hingga 20-50%, dan mortalitas hingga 5-20%.

c) ND (*Newcastle Disease*)

ND (*Newcastle Disease*) atau sering disebut tetelo merupakan suatu penyakit pernafasan sistemik yang bersifat akut dan mudah menular dengan tingkat kesakitan mencapai 80-100% dan angka kematian mencapai 100% yang menyebabkan penurunan produksi telur baik secara kualitas maupun kuantitas.

d) AI (*Avian Influenza*)

AI (*Avian Influenza*) merupakan penyakit virus yang menyerang sistem pernafasan, pencernaan, reproduksi, dan saraf pada berbagai spesies unggas. Penyakit ini

menyebabkan angka kesakitan dan kematian yang tinggi.

e) *Fowl Cholera*

*Fowl Cholera* adalah penyakit menular yang disebabkan oleh bakteri dan menyerang berbagai jenis unggas termasuk ayam. Penyakit ini ditandai dengan adanya kematian mendadak dengan angka kesakitan dan angka kematian mencapai 30-50%.

f) *Infectious Bursal Disease (IBD)*

*Infectious Bursal Disease (IBD)* atau disebut juga penyakit gumboro, penyakit ini umumnya menyerang ayam umur 0-8 minggu, penyakit ini menyerang sistem kekebalan tubuh pada ayam, ayam yang sudah terkena penyakit gumboro akan mudah terserang infeksi sekunder lainnya yang menyebabkan tingkat kematian mencapai 70% dan tingkat kesakitan mencapai 100%.

g) *Infectious Bronchitis (IB)*

*Infectious Bronchitis (IB)* merupakan penyakit viral yang menyerang saluran pernafasan dan sistem reproduksi ayam yang bersifat akut dan sangat mudah menular dengan masa inkubasi yang singkat, antara 18-

36 jam. Penyakit ini menyebabkan gangguan pertumbuhan, menurunkan kualitas daging (ayam pedaging) dan penurunan produksi telur.

h) *Koksidiosis*

*Koksidiosis* atau sering disebut berak darah adalah penyakit parasit protozoa yang menyerang saluran pencernaan bagian usus halus dan sekum. Penyakit ini menyebabkan penurunan kekebalan tubuh ayam sehingga rentan terkena infeksi penyakit lain.

**B. Kajian Penelitian yang Relevan**

Dibawah ini merupakan beberapa penelitian sebelumnya yang relevan berdasarkan kesamaan teknologi maupun topik yang menjadi referensi dari penelitian ini, yaitu sebagai berikut :

<b>1</b>	Nama peneliti & tahun	(Endra & Antika, 2021)
	Judul penelitian	Sistem Pakar menggunakan Metode Forward Chaining untuk Diagnosa Penyakit Tanaman Padi berbasis Android
	Tujuan penelitian	Penelitian ini bertujuan untuk membangun aplikasi sistem

		<p>pakar untuk membantu petani menyelesaikan masalah petani dalam mendiagnosa penyakit tanaman padi berbasis android.</p>
	Hasil penelitian	<p>Hasil dari penelitian ini adalah aplikasi sistem pakar dengan menggunakan Forward chaining dapat digunakan untuk diagnosa penyakit tanaman padi dengan beberapa langkah yaitu memilih gejala, melakukan konsultasi, dan menampilkan hasil konsultasi.</p>
2	Nama peneliti & tahun	<p>(Pasalli et al., 2016)</p>
	Judul penelitian	<p>Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining Berbasis Mobile</p>
	Tujuan penelitian	<p>Penelitian ini bertujuan untuk membuat aplikasi sistem pakar berbasis mobile dalam mendiagnosis penyakit anak dengan menggunakan teknik penalaran forward chaining</p>

	Hasil penelitian	Hasil dari penelitian ini adalah sebuah aplikasi sistem pakar berbasis mobile yang dapat mendiagnosa 9 penyakit pada anak.
3	Nama peneliti & tahun	(Putra et al., 2019)
	Judul penelitian	Sistem Pakar Diagnosis Penyakit Ginjal Dengan Metoda Forward Chaining
	Tujuan penelitian	Penelitian ini bertujuan untuk membuat sebuah sistem Pakar yang dapat digunakan sebagai alat bantu dalam mendiagnosa penyakit ginjal.
	Hasil penelitian	Hasil dari penelitian ini adalah sebuah sistem yang dapat menentukan penyakit ginjal dari gejala-gejala yang dirasakan dengan cepat dan tepat sehingga penelitian ini sangat membantu dalam mengurangi resiko keterlambatan diagnosis penyakit ginjal.

4	Nama peneliti & tahun	(Febronius Bere et al., 2021)
	Judul penelitian	Sistem Pakar Diagnosis Penyakit Pada Ayam Menggunakan Metode <i>Certainly Factor</i>
	Tujuan penelitian	Tujuan penellitian ini adalah merancang sebuah sistem pakar yang dapat digunakan untuk mendiagnosis penyakit pada ayam menggunakan metode <i>certainly factor</i>
	Hasil penelitian	Hasil dari penelitian ini adalah sistem pakar yang memberikan informasi kepada peternak ayam mengenai penyakit pada ayam, berdasarkan gejala yang timbul akibat serangan penyakit, dan sitem ini juga dapat menampilkan solusi penanganan untuk ayam yang sedang terserang penyakit serta menampilkan nilai presentasi metode <i>certainly factor</i> .

## **BAB III**

### **METODE PENELITIAN**

#### **A. Metode Pengumpulan Data**

##### 1. Wawancara

Penulis akan melakukan pengumpulan data yang berbentuk tanya jawab. Wawancara dilakukan penulis kepada narasumber untuk mendapatkan data yang diharapkan. Penulis bertindak sebagai pewawancara dan narasumber bertindak sebagai informan.

##### 2. Studi Pustaka

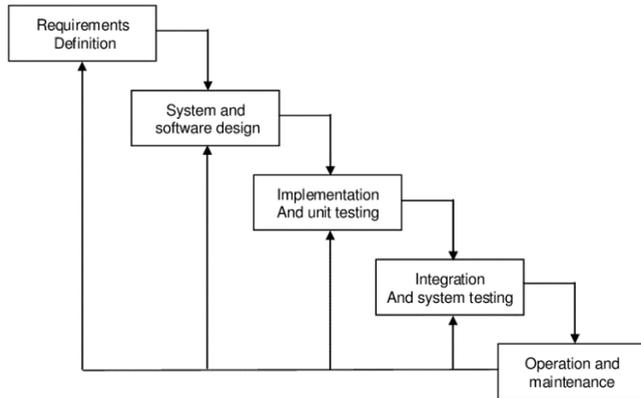
Studi pustaka dilakukan untuk mendapatkan referensi tentang teori-teori yang diperlukan dengan cara mengumpulkan dan membaca berbagai macam literatur yang berkaitan dengan penelitian yang bersumber dari jurnal, skripsi, maupun artikel yang beredar di internet.

#### **B. Metode Pengembangan Sistem**

##### 1. *Waterfall*

Metode pengembangan yang digunakan dalam penelitian ini adalah metode *waterfall*, yaitu metode pengembangan yang menekankan pada fase yang berurutan dan sistematis seperti air terjun. Metode *waterfall* memiliki 5 tahapan, yaitu *requirement analisyst and definition, system and software design, implementation and unit testing, integration and*

*system testing*, dan *operation and maintenance* (Sommerville, 2011).



Gambar 1.0 Tahapan Metode *Waterfall*

Berikut adalah tahapan-tahapannya :

a) *Requirement Analysis*

Tahap pertama adalah *Requirement Analysis*, yaitu pengembang harus mengetahui seluruh informasi tentang kebutuhan software seperti fungsi software yang diperlukan oleh pengguna dan juga batasan software. Informasi tersebut bisa diperoleh melalui wawancara, survey, ataupun diskusi. Informasi yang telah diperoleh kemudian dianalisis sehingga didapatkan data-data mengenai kebutuhan pengguna terhadap software yang akan dikembangkan. Dalam penelitian ini sistem pakar akan berbentuk

website yang dibangun dengan metode pengembangan *waterfall* dan Bahasa pemrograman PHP.

b) *System Design*

Tahap kedua yaitu *System Design*. *Design* dilakukan sebelum proses *coding* dimulai. Tujuan dari design adalah untuk memberikan gambaran apa saja sesuatu yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem yang akan dibuat sehingga kebutuhan hardware dan sistem dapat lebih spesifik, design juga berfungsi untuk mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan.

1) Data Flow Diagram (DFD)

DFD (Data Flow Diagram) merupakan suatu permodelan untuk menggambarkan sebuah sistem sebagai suatu jaringan fungsional yang dihubungkan dengan manual maupun komputerisasi antara satu sama lain, DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan

pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur (A.S & Salahudin, 2016).

## 2) *Wireframe*

*Wireframing* merupakan tahapan penting dalam proses merancang sebuah media digital (Segara, 2019). *Wireframe* merupakan sebuah kerangka untuk menata suatu item di laman website atau aplikasi. Pembuatan *wireframe* biasanya dilakukan sebelum pembuatan produk tersebut dilakukan. Item yang berkaitan seperti teks, gambar, dan sebagainya.

## 3) Kerangka Sistem Pakar

Kerangka Sistem pakar berfungsi untuk memperlihatkan cara kerja sistem pakar secara sederhana jika dilihat dari proses input proses dan output.

## 4) Flowchart

Flowchart berfungsi sebagai gambaran sebuah alur pengerjaan maupun proses suatu program, di mana proses tersebut digambarkan dalam bentuk flowchart simbol yang praktis dan mudah untuk dipahami.

c) *Implementation*

Tahap ketiga adalah *implementation*, yaitu proses coding yang akan dipecah menjadi modul-modul kecil yang kemudian nanti akan digabungkan dalam tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah berfungsi seperti yang diinginkan atau belum.

d) *Integration and Testing*

Tahap keempat adalah *Integration and Testing*, yaitu proses penggabungan modul-modul yang sudah dibuat sebelumnya kemudian dilakukan pengujian trial error yang bertujuan untuk mengetahui apakah software sudah sesuai desain yang diinginkan dan apakah masih terdapat error atau tidak. Dalam penelitian ini menggunakan metode black box.

e) *Operation and Maintenance*

Tahap terakhir dalam metode *waterfall* adalah *Operation and Maintenance*. Pada tahap ini software yang sudah jadi tinggal dijalankan atau dioperasikan oleh penggunanya. Disamping itu juga dilakukan pemeliharaan termasuk perbaikan kesalahan, perbaikan sistem, dan juga

peningkatan fungsi sistem sesuai kebutuhan yang baru.

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### A. Requirement Analysis

Pada proses *Requirement Analysis* didapatkan informasi kebutuhan pengguna dan tabel keputusan yang dibutuhkan dalam sistem pakar berbasis web yang bersumber dari wawancara dan juga studi pustaka dari jurnal penelitian sebelumnya.

##### 1. Kebutuhan Sistem

No	Penguuna	Kebutuhan
1	Pakar/admin	Login ke dalam website
2	Pakar/admin	Menambah data penyakit
3	Pakar/admin	Mengedit data penyakit
4	Pakar/admin	Menghapus data penyakit
5	User	Registrasi akun
6	User	Login ke dalam website
7	User	Menginputkan gejala untuk cek penyakit

Tabel 1.0 Tabel Kebutuhan Sistem

Hasil wawancara dengan Bapak Diar selaku pemilik ternak pribadi terkait penyakit yang pernah menyerang ternaknya adalah AI(avian influenza) dengan gejala yang diingat adalah ayam nafsu makannya berkurang, jengger berwarna biru, cekreh, dan juga produksi telur menurun, namun karena minimnya informasi terkait gejala yang dialami yang disebabkan tidak ada pencatatan data,

sehingga penulis mewawancarai Bapak Andri Setiawan S.Pt, salah satu tenaga ahli dari ternak Agrosari Farm untuk mendapatkan data penyakit yang pernah menyerang ternak Agrosari farm, didapatkan data penyakit sebagai berikut :

Penyakit	: Coriza/SNOT
Gejala	- Nafsu makan turun
	- Ada leleran seperti nanah dimata
	- Ayam cekreh, bersin
	- Bengkak pada mata dan hidung
	- Produksi turun

Tabel 2.0 Tabel Data Penyakit Hasil Wawancara

Data penyakit diatas digunakan untuk melakukan pengujian sistem apakah sudah bisa berjalan atau belum. Kemudian didapatkan data pengetahuan dari jurnal penelitian sebelumnya sebagai berikut :

2. Tabel Basis Pengetahuan

**Tabel Penyakit**

<b>Daftar Penyakit</b>	
<b>Kode</b>	<b>Nama Penyakit</b>
P1	Egg Drop Syndrome 76/EDS 76
P2	<i>Infectious Coryza</i> (SNOT)
P3	ND ( <i>Newcastle Disease</i> )
P4	AI ( <i>Avian Influenza</i> )
P5	<i>Infectious Bronchitis</i> (IB)
P6	Koksidiosis
P7	<i>Infectious Bursal Disease</i> (IBD)
P8	<i>Fowl Cholera</i>

Tabel 3.1 Tabel Penyakit

## Tabel Gejala

<b>Daftar Gejala</b>	
<b>Kode</b>	<b>Gejala</b>
G1	Diare
G2	Nafas sesak/megap-megap
G3	Nafas ngorok
G4	Nafas cepat
G5	Bersin-bersin
G6	Batuk
G7	Badan kurus
G8	Bulu kusam dan berkerut
G9	Nafsu makan berkurang
G10	Produksi telur menurun
G11	Kualitas telur jelek
G12	Kelihatan ngantuk dan bulu berdiri
G13	Kedinginan
G14	Tampak lesu
G15	Mencret kehijau-hijauan
G16	Mencret keputih-putihan
G17	Mencret bercampur darah
G18	Banyak minum
G19	Muka pucat
G20	Nampak membiru

G21	Sempoyongan
G22	Jengger membengkak merah
G23	Kaki meradang/lumpuh
G24	Kelopak mata kemerahan
G25	Keluar cairan berbusa dari mata
G26	Keluar cairan dari mata dan hidung
G27	Keluar nanah dari mata dan bau
G28	Kepala bengkak
G29	Kepala terputar
G30	Pembengkakan dari sinus dan mata
G31	Tidur paruhnya diletakkan dilantai
G32	Duduk dengan skap membungkuk
G33	Mati secara mendadak

Tabel 3.2 Tabel Gejala

**Tabel Keputusan**

<b>Kode</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>
<b>G1</b>		<b>X</b>		<b>X</b>	<b>X</b>			<b>X</b>
<b>G2</b>			<b>X</b>	<b>X</b>				<b>X</b>
<b>G3</b>			<b>X</b>	<b>X</b>	<b>X</b>			<b>X</b>
<b>G4</b>	<b>X</b>							
<b>G5</b>		<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>			
<b>G6</b>			<b>X</b>	<b>X</b>	<b>X</b>			
<b>G7</b>						<b>X</b>		
<b>G8</b>						<b>X</b>	<b>X</b>	<b>X</b>
<b>G9</b>		<b>X</b>						
<b>G10</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>		<b>X</b>
<b>G11</b>	<b>X</b>							
<b>G12</b>					<b>X</b>			
<b>G13</b>					<b>X</b>			
<b>G14</b>			<b>X</b>		<b>X</b>		<b>X</b>	
<b>G15</b>	<b>X</b>		<b>X</b>					<b>X</b>
<b>G16</b>							<b>X</b>	
<b>G17</b>						<b>X</b>		
<b>G18</b>								<b>X</b>
<b>G19</b>						<b>X</b>		
<b>G20</b>				<b>X</b>	<b>X</b>			
<b>G21</b>			<b>X</b>					

<b>G22</b>								<b>X</b>
<b>G23</b>								<b>X</b>
<b>G24</b>		<b>X</b>						
<b>G25</b>				<b>X</b>				
<b>G26</b>								<b>X</b>
<b>G27</b>		<b>X</b>						
<b>G28</b>				<b>X</b>				
<b>G29</b>			<b>X</b>					
<b>G30</b>		<b>X</b>						
<b>G31</b>							<b>X</b>	
<b>G32</b>							<b>X</b>	
<b>G33</b>				<b>X</b>				

Tabel 3.3 Tabel relasi penyakit dan gejala (Tentua, 2009)

## **Aturan Sistem Pakar**

### **Rule 1**

- If nafas cepat,
- And produksi telur menurun,
- And kualitas telur jelek,
- And mencret kehijau-hijauan,
- Then *Egg Drop Syndrome 76/EDS 76*

### **Rule 2**

- If diare,
- And bersin-bersin,
- And nafsu makan berkurang,
- And produksi telur menurun,
- And kelopak mata kemerahan,
- And keluar nanah dari mata dan bau.
- And pembengkakan pada sinus dan mata
- Then *Infectious Coryza(SNOT)*

### **Rule 3**

- If nafsu makan berkurang,
- And nafas sesak/megap-megap,
- And nafas ngorok,
- And bersin-bersin,
- And batuk,
- And produksi telur menurun,
- And tampak lesu,

- And mencret kehijau-hijauan.
- And sempoyongan,
- And kepala berputar,
- Then ND (*Newcastle Disease*)

#### **Rule 4**

- If diare,
- And nafas sesak/megap-megap,
- And nafas ngorok,
- And bersin-bersin,
- And batuk,
- And nafsu makan berkurang,
- And produksi telur menurun,
- And nampak membiru,
- And keluar cairan berbusa dari mata,
- And kepala bengkak,
- And mati secara mendadak.
- Then AI (*Avian Influenza*)

#### **Rule 5**

- If diare,
- And nafas ngorok,
- And bersin-bersin,
- And batuk,
- And nafsu makan berkurang,
- And produksi telur menurun,

- And kelihatan mengantuk dengan bulu berdiri.
- And kedinginan,
- And tampak lesu,
- And nampak membiru,
- Then *Infectious Bronchitis* (IB)

#### **Rule 6**

- If nafsu makan berkurang,
- And badan kurus,
- And bulu kusan dan berkerut,
- And produksi telur menurun,
- And muka pucat,
- And mencret bercampur darah
- Then Koksidiosis

#### **Rule 7**

- If nafsu makan berkurang,
- And bulu kusam dan berkerut,
- And tampak lesu,
- And mencret keputih-putihan,
- And tidur paruhnya diletakkan dilantai,
- And duduk dengan sikap membungkuk,
- Then *Infectious Bursal Disease* (IBD)

#### **Rule 8**

- If diare,
- And nafas sesak/megap-megap,

- And nafas ngorok,
- And bulu kusam dan berkerut,
- And nafsu makan berkurang,
- And produksi telur menurun,
- And mencret kehijau-hijauan,
- And banyak minum,
- And jengger membengkak merah,
- And kaki meradang/lumpuh,
- And keluar cairan dari mata dan hidung.
- Then *Fowl Cholera*

## **B. System Design**

Pada proses *System Design* terdapat beberapa gambaran apa saja yang akan dilakukan sebelum proses coding dimulai yang menghasilkan 4 gambaran, yang pertama yaitu Data Flow Diagram yang berfungsi sebagai representasi bagaimana sebuah sistem pakar berbasis web ini bekerja. Lalu ada Wireframe sebagai kerangka untuk menata suatu item dilaman *website* berkaitan dengan teks, gambar maupun yang lainnya. Kemudian ada kerangka sistem pakar yang memperlihatkan cara kerja sistem pakar secara sederhana jika dilihat dari proses input proses dan output. Dan yang terakhir ada Flowchart untuk

menggambarkan bagaimana alur sistem pakar berbasis web ini bekerja.

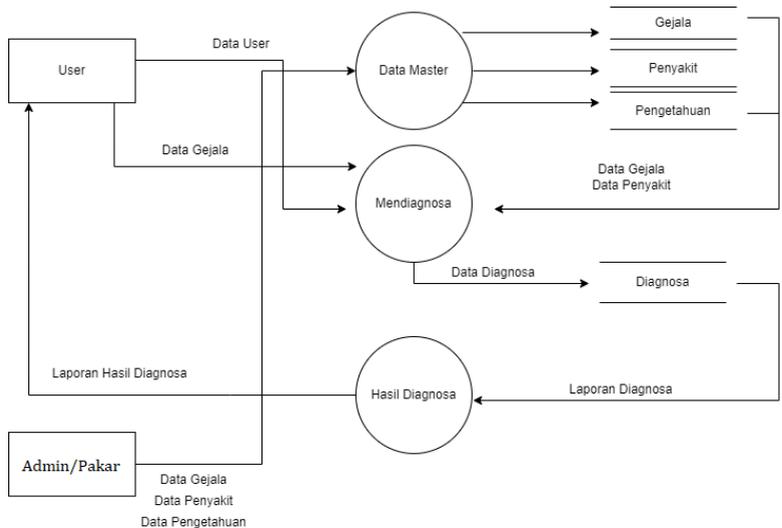
## 1. Data Flow Diagram

### Diagram Konteks



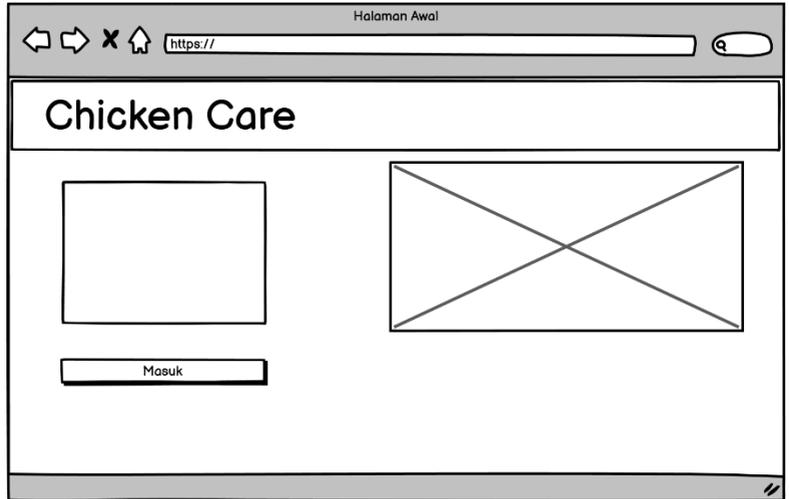
Gambar 2.1 Diagram Konteks

### Data Flow Diagram Level 0

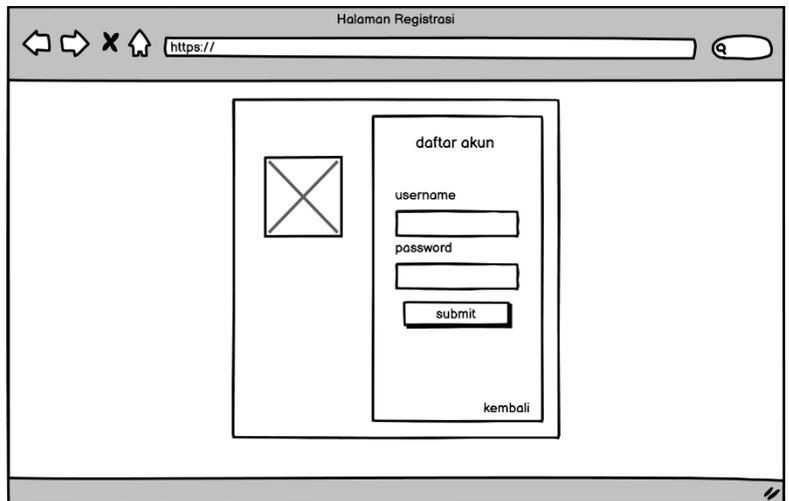


Gambar 2.2 Data Flow Diagram Level 0

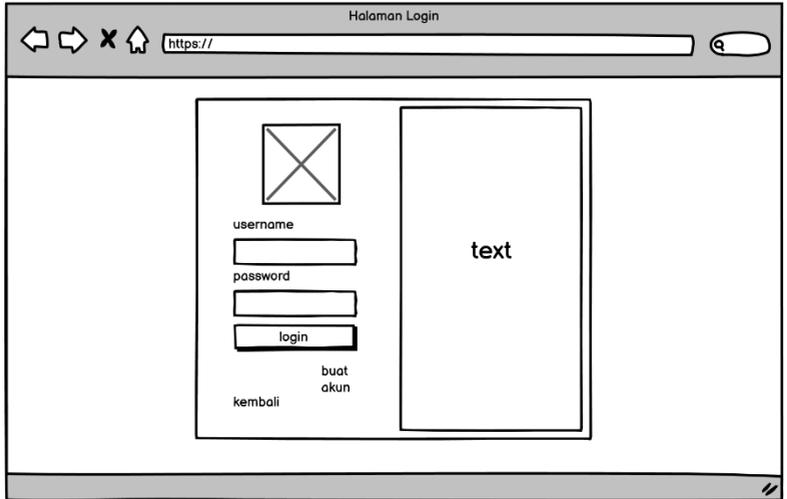
## 2. Wireframe



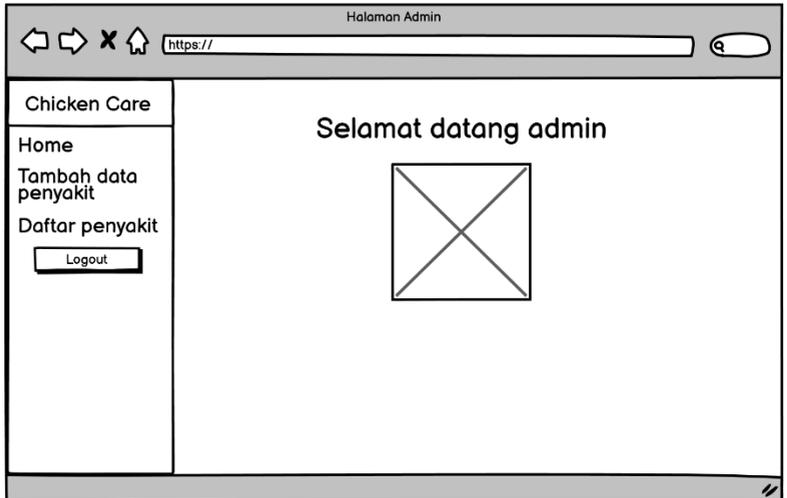
Gambar 3.1 Wireframe Halaman Awal Website



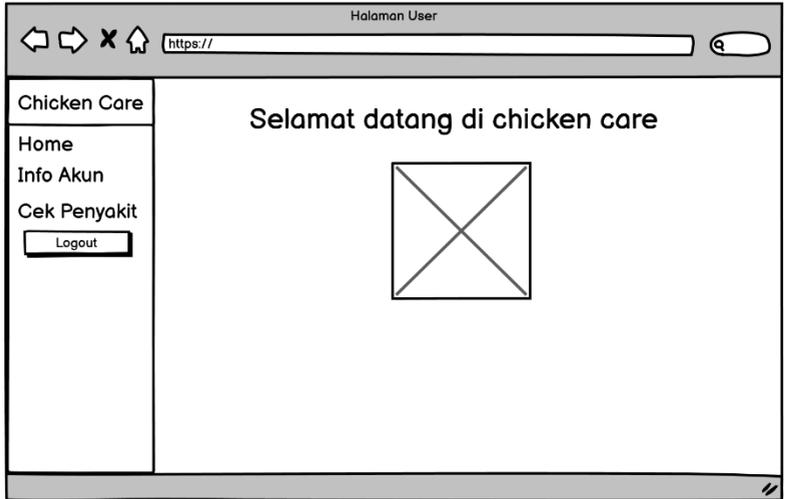
Gambar 3.2 Wireframe Halaman Daftar Akun



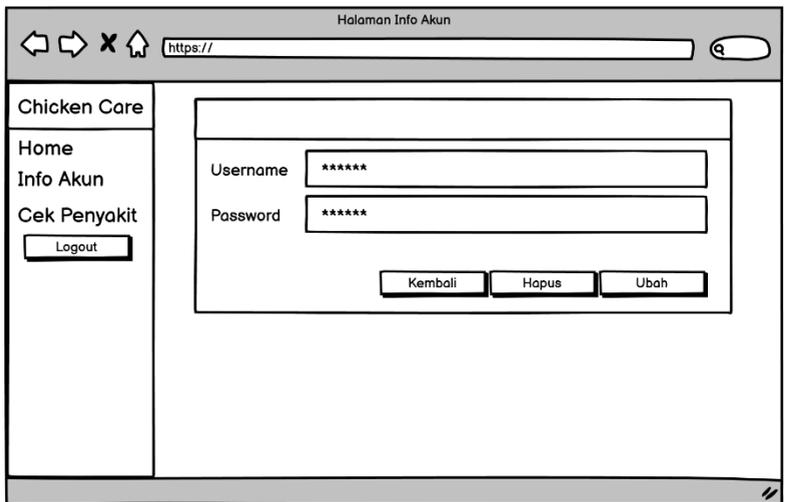
Gambar 3.3 *Wireframe* Halaman Login



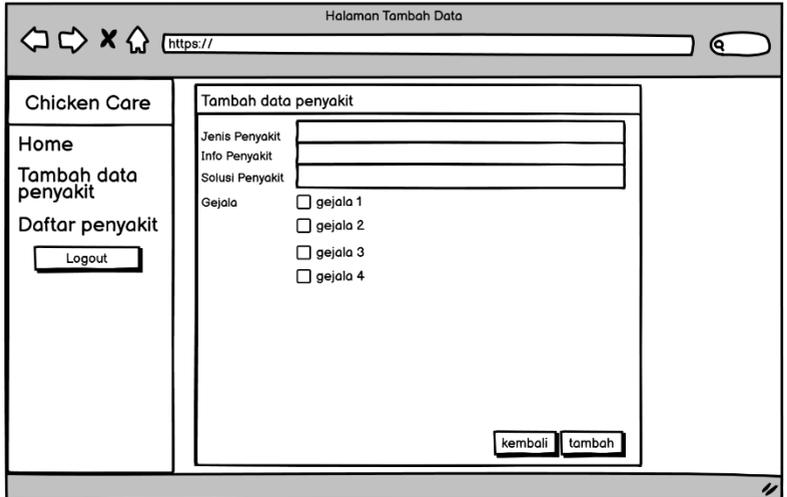
Gambar 3.4 *Wireframe* Halaman Admin



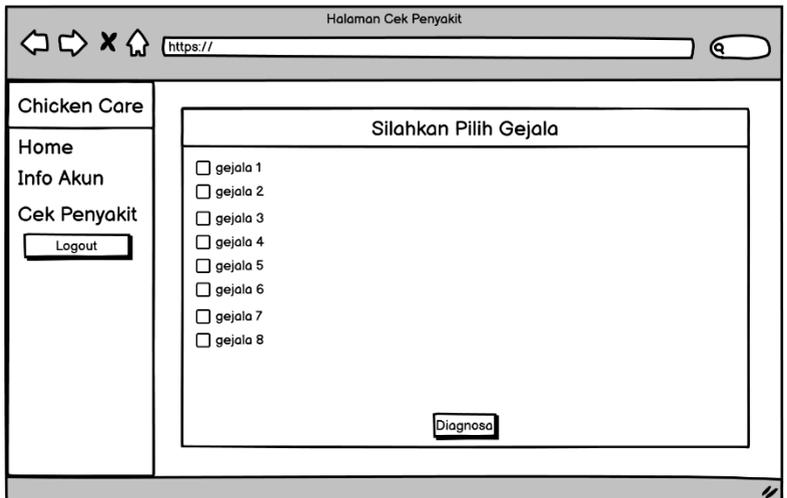
Gambar 3.5 *Wireframe* Halaman User



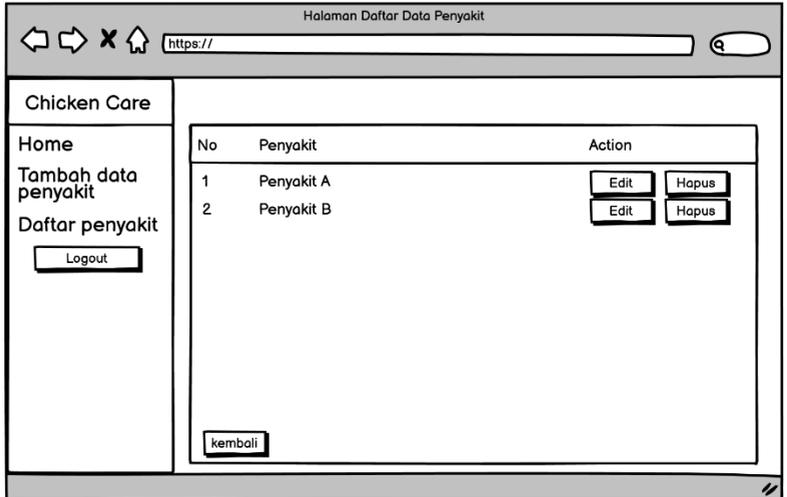
Gambar 3.6 *Wireframe* Halaman Info Akun



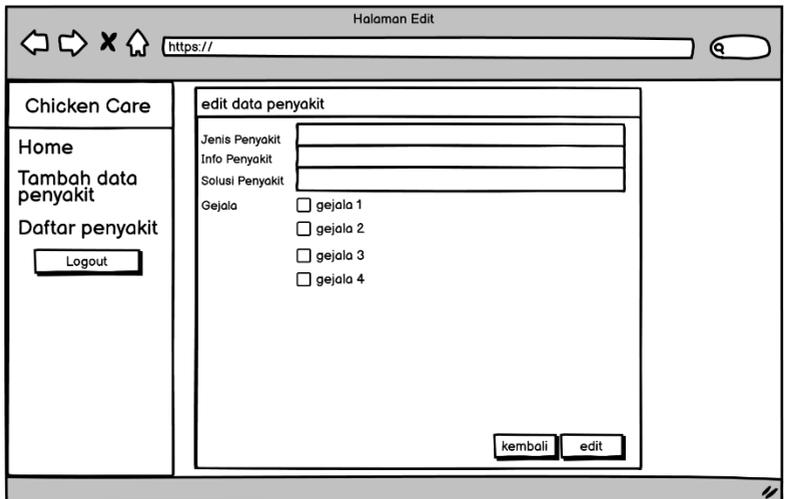
Gambar 3.7 Wireframe Halaman Tambah Data Penyakit



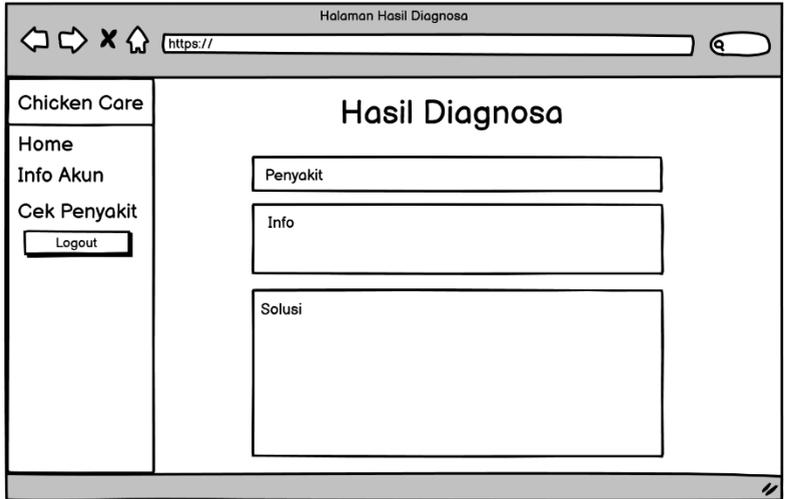
Gambar 3.8 Wireframe Halaman Cek Penyakit



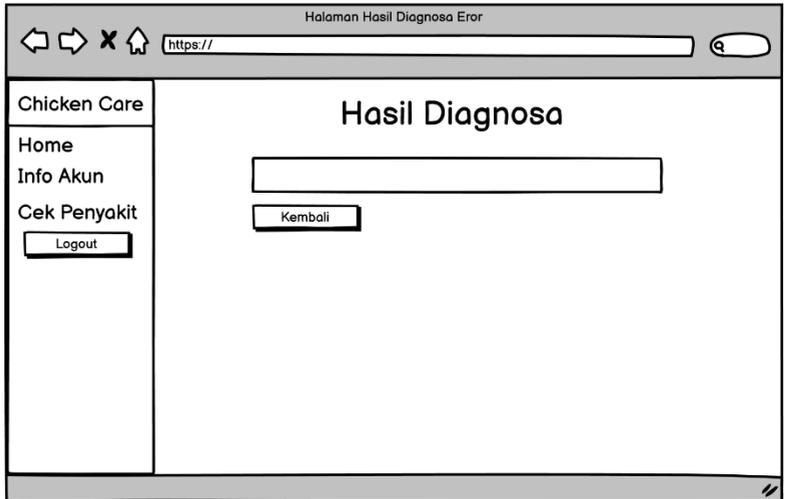
Gambar 3.9 Wireframe Halaman Daftar Penyakit



Gambar 3.10 Wireframe Halaman Edit Data Penyakit



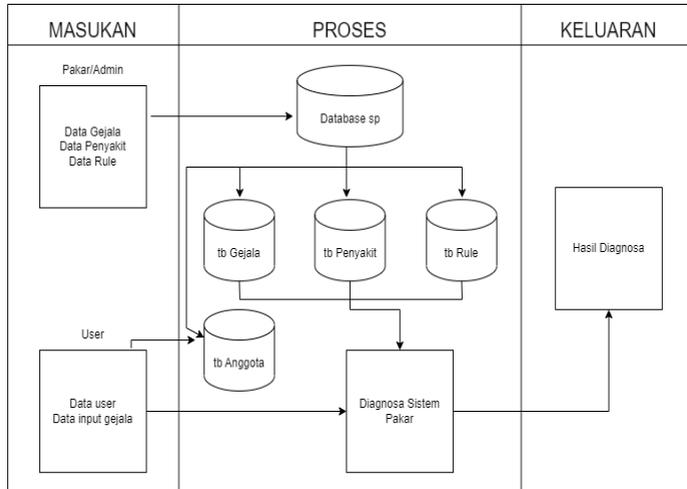
Gambar 3.11 *Wireframe* Halaman Hasil Diagnosa



Gambar 3.12 *Wireframe* Halaman Hasil Error

Diatas merupakan beberapa hasil dari *wireframe* yang kurang lebih akan digunakan didalam website sistem pakar dalam penelitian ini.

### 3. Kerangka Sistem Pakar



Gambar 4.0 Kerangka Sistem Pakar

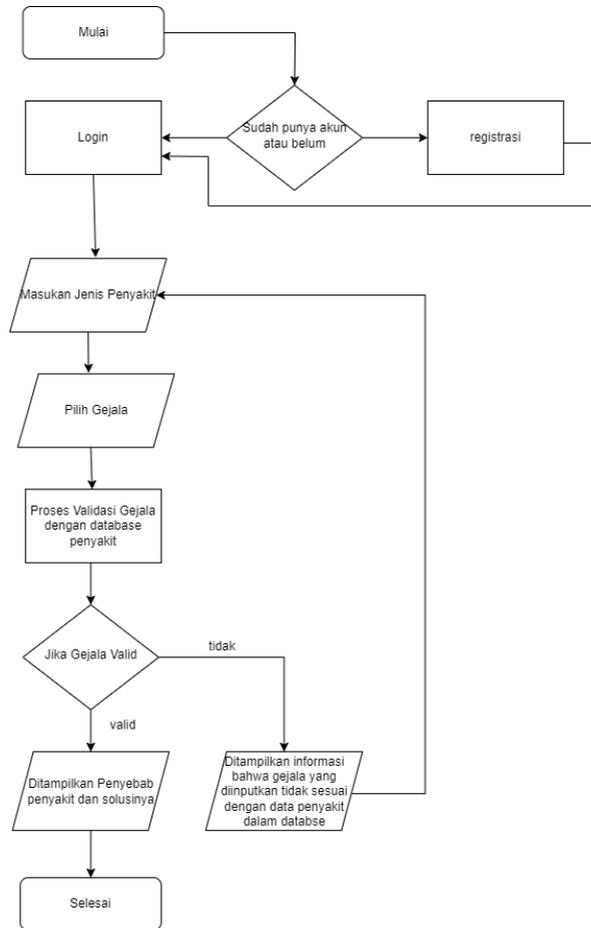
Pada tahapan masukan, pakar/admin memasukan data penyakit, data gejala, dan data pengetahuan, sedangkan *user* memasukan data user dan menginputkan gejala yang terjadi.

Pada tahapan proses, data yang dimasukkan oleh pakar/admin akan disimpan didalam database yang berisi tabel gejala, tabel penyakit, dan tabel rule, kemudian pada tahapan proses ini, data user akan disimpan didalam database di tabel anggota agar bisa

melakukan proses login, sedangkan data gejala yang diinputkan *user* akan diolah didalam sistem dengan menggunakan metode *forward chaining* dengan cara mencocokkan satu persatu antara data gejala yang diinputkan dengan data dalam tabel rule yang terdapat didalam database secara berulang.

Pada tahapan keluaran, jika data gejala yang diinputkan user sesuai dengan data didalam tabel rule, maka sistem akan memberikan output berupa jenis penyakit, penyebab penyakit, solusi pencegahan, penanganan, dan perawatan.

#### 4. Flowchart



Gambar 5.0 *Flowchart*

Sistem dimulai dengan *user* disuruh untuk login terlebih dahulu, jika belum mempunyai akun *user* bisa registrasi terlebih dahulu. Kemudian setelah registrasi berhasil, kemudian login ke dalam website sistem pakar,

setelah itu *user* disuruh memilih gejala yang ditampilkan oleh sistem secara berurutan sampai selesai. Setelah itu sistem akan memproses apakah gejala yang dimasukkan sesuai didalam rule penyakit yang ada, valid atau tidak. Jika gejala valid, maka sistem akan menampilkan apa jenis penyakit, info penyakit, juga solusi penanganan dan perawatannya kemudian sistem akan selesai. Tetapi jika gejala yang dimasukkan tidak valid, maka sistem akan menampilkan informasi bahwa gejala yang dimasukkan tidak sesuai dengan data didalam rule penyakit yang ada didalam database dan akan dikembalikan ke menu pemilihan gejala.

### C. *Implementation*

Pada proses *Implementation* dilakukan proses *coding* dari setiap bagian dengan hasil sebagai berikut :

#### a. *Proses Coding*

##### *Coding* untuk login

```
1 <?php
2 session_start();
3 $nama = $_POST['nama'];
4 $password = $_POST['password'];
5
6 $conn = mysqli_connect("localhost", "root", "", "db_sp");
7 $query = "SELECT * from anggota where nama='$nama' && password='$password'";
8 $login = mysqli_query($conn, $query);
9 $islogin = mysqli_num_rows($login);
10
11 if (isset($_POST['tblogin'])) {
12     if ($islogin > 0) {
13         $am = mysqli_fetch_assoc($login);
14         $_SESSION["nama"] = $am["nama"];
15         $_SESSION["password"] = $am["password"];
16         $_SESSION["level"] = $am["level"];
17         $_SESSION['login'] = true;
```

Gambar 6.1 *Coding* Proses Login (1)

Pada proses login, Memulai sesi dengan *session\_start()*, lalu mengambil *value* nama lalu disimpan dalam variabel *\$nama* dan *value password* disimpan dalam variabel *\$password* menggunakan *\$\_POST*. Kemudian membuat koneksi dengan *database* dengan nama *db\_sp* menggunakan *mysqli\_connect*, lalu membuat kueri *SQL* untuk menyeleksi data dalam tabel anggota dengan kondisi dimana nama = *\$nama* dan *password* = *\$password*. Kueri dieksekusi kemudian ditampung dalam variable *\$login* lalu memeriksa jumlah baris yang dipanggil dari *database* hasil eksekusi *mysqli\_query* dengan menggunakan *mysqli\_num\_rows(\$login)* dan menyimpan hasilnya dalam bentuk numerik didalam variabel *\$islogin*. Kemudian mengecek apakah tombol *tblogin* telah ditekan dengan *isset(\$\_POST['tblogin'])*; Jika *\$islogin > 0*, maka ditemukan data yang cocok didalam *database*, kemudian mengambil data didalam variable *\$login* menggunakan *mysqli\_fetch\_assoc(\$login)* dan menyimpannya dalam variabel *\$am*. Variabel *\$\_SESSION* berisi nama, *password*, level, dan id yang diambil dari variabel *\$am*. Setelah itu variabel *\$\_SESSION['login']* menjadi bernilai *true* yang menunjukkan jika sudah berhasil masuk

```

18
19     if ($_SESSION["level"] == 0) {
20         header("Location: user.php");
21     } else if ($_SESSION["level"] == 1) {
22         header("Location: admin.php");
23     }
24 } else {
25     $msg = "<p class='alert alert-danger' roles='alert'>username atau password salah, mohon memasukkan username atau password
26         dengan benar </p>";
27     header("Location: login.php?msg=$msg");
28 } else {
29     echo "tombol tidak ditekan";
30 }

```

Gambar 6.2 Coding Proses Login (2)

Jika level *user* adalah 0, maka akan diarahkan ke halaman *user.php* dan jika level *user* adalah 1, maka akan diarahkan ke halaman *admin.php*. kembali lagi ke atas, jika *\$islogin* memiliki nilai 0 berarti tidak ada data yang cocok didalam database dan akan diarahkan ke halaman login dengan pesan “*username* atau *password* salah, mohon masukkan *username* atau *password* dengan benar”.

## Coding untuk registrasi/buat akun

```

1 <?php
2 session_start();
3 $conn = mysqli_connect("localhost", "root", "", "db_sp");
4
5 if (isset($_POST['submit-akn'])) {
6     $nama = $_POST['nama'];
7     $password = $_POST['password'];
8
9     $query = "insert into anggota(nama,password,level) values('$nama','$password','0')";
10
11     if (mysqli_query($conn, $query) == 'true') {
12         $message = "Selamat datang di chicken care dan silahkan login";
13     } else {
14         $message = "Maaf coba ulangi lagi" . mysqli_error($conn);
15     }
16 }
17 ?>

```

Gambar 6.3 Coding Proses Registrasi

Memulai sesi PHP dengan `session_start();`, kemudian membuat koneksi dengan `database db_sp` dengan `mysqli_connect`. Lalu memeriksa apakah tombol `submit-mnk` sudah ditekan pada menu buat akun dengan `isset($_POST['submit-akn'])`.

```

</div>
<div class="container-right">
<div class="title title-right">Daftar Akun</div>
<form action="proses-sign-up.php" method="post">
<div class="form-outline mb-4">
<label class="form-label" for="nama">Nama</label>
<input type="text" id="nama" required name="nama" class="form-control" />
</div>
<div class="form-outline mb-4">
<label class="form-label" for="password">Password</label>
<input type="password" id="password" required name="password" class="form-control" />
</div>
<button type="submit" name="submit-akn" class="btn ml-5">Submit</button>
</form>

```

Gambar 6.4 Coding tombol `submit-akn`

Lalu mengambil `value` nama lalu disimpan dalam variabel `$nama` dan `value password` disimpan dalam variabel `$password` menggunakan `$_POST`. Kemudian membuat kueri `SQL` untuk menambahkan data kedalam `database` tabel anggota dengan nama yang memiliki `value $nama`, `password` dengan `value $password`, dan level dengan `value 0`. Lalu eksekusi kueri dengan `mysqli_query == true`, jika berhasil maka akan muncul pesan selamat datang di chicken care dan silahkan login, jika gagal maka muncul pesan maaf coba ulang lagi.

## Coding untuk menambahkan data penyakit

```
1 <?php
2 $conn = mysqli_connect("localhost", "root", "", "db_sp");
3
4 if (isset($_POST['submit-gjl'])) {
5     $penyakit = $_POST['penyakit'];
6     $info = $_POST['info'];
7     $solusi = $_POST['solusi'];
8     $G001 = isset($_POST['G001']) ? 1 : 0;
9     $G002 = isset($_POST['G002']) ? 1 : 0;
10    $G003 = isset($_POST['G003']) ? 1 : 0;
11    $G004 = isset($_POST['G004']) ? 1 : 0;
12    $G005 = isset($_POST['G005']) ? 1 : 0;
13    $G006 = isset($_POST['G006']) ? 1 : 0;
14    $G007 = isset($_POST['G007']) ? 1 : 0;
15    $G008 = isset($_POST['G008']) ? 1 : 0;
16    $G009 = isset($_POST['G009']) ? 1 : 0;
17    $G010 = isset($_POST['G010']) ? 1 : 0;
18    $G011 = isset($_POST['G011']) ? 1 : 0;
19    $G012 = isset($_POST['G012']) ? 1 : 0;
20    $G013 = isset($_POST['G013']) ? 1 : 0;
21    $G014 = isset($_POST['G014']) ? 1 : 0;
22    $G015 = isset($_POST['G015']) ? 1 : 0;
23    $G016 = isset($_POST['G016']) ? 1 : 0;
24    $G017 = isset($_POST['G017']) ? 1 : 0;
25    $G018 = isset($_POST['G018']) ? 1 : 0;
26    $G019 = isset($_POST['G019']) ? 1 : 0;
```

Gambar 6.5 Coding Tambah Data Penyakit (1)

```
27    $G020 = isset($_POST['G020']) ? 1 : 0;
28    $G021 = isset($_POST['G021']) ? 1 : 0;
29    $G022 = isset($_POST['G022']) ? 1 : 0;
30    $G023 = isset($_POST['G023']) ? 1 : 0;
31    $G024 = isset($_POST['G024']) ? 1 : 0;
32    $G025 = isset($_POST['G025']) ? 1 : 0;
33    $G026 = isset($_POST['G026']) ? 1 : 0;
34    $G027 = isset($_POST['G027']) ? 1 : 0;
35    $G028 = isset($_POST['G028']) ? 1 : 0;
36    $G029 = isset($_POST['G029']) ? 1 : 0;
37    $G030 = isset($_POST['G030']) ? 1 : 0;
38    $G031 = isset($_POST['G031']) ? 1 : 0;
39    $G032 = isset($_POST['G032']) ? 1 : 0;
40    $G033 = isset($_POST['G033']) ? 1 : 0;
```

Gambar 6.6 Coding Tambah Data Penyakit (2)

Pada proses tambah data, langkah pertama lakukan koneksi dengan *database* yang memiliki nama

db\_sp dengan *mysqli\_connect*, kemudian isset *\$\_POST submit-gjl* untuk mengecek tombol *submit-gjl* pada halaman tambah data ditekan atau tidak, jika ditekan maka akan mengambil *value* dari penyakit yang disimpan dalam variabel *\$penyakit*, *value* info disimpan didalam variabel *\$info*, *value* solusi disimpan dalam variabel *\$solusi*. Kemudian *value* G001 – G033 yang disimpan didalam variabel *\$G001 - \$G033* yang menunjukkan *value* 1 apabila *radiobox* dicentang dan *value* 0 apabila *radiobox* tidak dicentang.

```

42 $sql = "insert into tb_penyakit (penyakit,info,solusi) values('$penyakit','$info','$solusi')";
43 $sql = "insert into tb_rule (G001,G002,G003,G004,G005,G006,G007,G008,G009,G010,G011,G012,G013,G014,G015,G016,G017,G018,G019,G020,
G021,G022,G023,G024,G025,G026,G027,G028,G029,G030,G031,G032,G033) values('$G001','$G002','$G003','$G004','$G005','$G006','$G007',
'$G008','$G009','$G010','$G011','$G012','$G013','$G014','$G015','$G016','$G017','$G018','$G019','$G020','$G021','$G022','$G023',
'$G024','$G025','$G026','$G027','$G028','$G029','$G030','$G031','$G032','$G033')";
44
45 if (mysqli_multi_query($conn, $sql)) {
46     $message = "Data penyakit berhasil ditambahkan";
47 } else {
48     $message = "Data penyakit gagal ditambahkan" . mysqli_error($conn);
49 }

```

Gambar 6.7 Coding Tambah Data Penyakit (3)

Kemudian membuat dua kueri *SQL* dengan *insert into* *tb\_penyakit* dan *insert into* *tb\_rule* untuk memasukan data kedalam tabel *tb\_penyakit* dan juga *tb\_rule* dengan *value* yang telah didapat dari proses sebelumnya. Lalu eksekusi kueri menggunakan *mysqli\_multi\_query*, jika berhasil maka data didalam table *tb\_penyakit* dan *tb\_rule* akan ditambah dan akan menampilkan pesan data penyakit telah berhasil ditambahkan, dan jika tidak berhasil maka data

didalam table `tb_penyakit` dan `tb_rule` tidak akan ditambah akan menampilkan pesan data penyakit gagal ditambahkan.

### **Coding** untuk menghapus data penyakit

```
<td><?=$data['penyakit'] ?></td>
<td><a href="edit-data.php?id<?=$data['id'] ?>" class="badge badge-primary">Edit</a>
<a href="aksi.php?del&id<?=$data['id'] ?>" class="badge badge-warning">Delete</a></td>
```

Gambar 6.8 *Coding URL* dengan parameter del

Pada proses hapus data, menggunakan *isset \$\_GET* untuk mengambil *URL* yang berisi parameter del berdasarkan id ketika tombol *delete* ditekan pada menu hapus data.

```
50 } else if (isset($_GET['del'])) {
51     $id = $_GET['id'];
52
53     $sql = "delete from tb_penyakit where id=$id;";
54     $sql .= "delete from tb_rule where id=$id;";
55     if (mysqli_multi_query($conn, $sql)) {
56         $message = "Data penyakit berhasil dihapus";
57     } else {
58         $message = "Data penyakit tidak terhapus" . mysqli_error($conn);
59     }
}
```

Gambar 6.9 *Coding Hapus Data Penyakit*

Kemudian membuat dua kueri dengan *delete from* `tb_penyakit` *where id=\$id* dan *delete from* `tb_rule` *where id=\$id* untuk menghapus data didalam tabel `tb_penyakit` dan data didalam tabel `tb_rule` berdasarkan id yang diambil dari parameter del. Lalu eksekusi kueri menggunakan *mysqli\_multi\_query*, jika

berhasil maka data didalam tabel tb\_penyakit dan data didalam tabel tb\_rule akan terhapus dan akan muncul pesan data penyakit berhasil dihapus, dan jika tidak berhasil maka data didalam tabel tb\_penyakit dan data didalam tabel tb\_rule tidak akan terhapus akan muncul pesan data penyakit tidak terhapus.

### **Coding** untuk menghapus data penyakit

```
60 } else if (isset($_POST['edit-gjl'])) {
61     $id = $_POST['id'];
62     $penyakit = $_POST['penyakit'];
63     $info = $_POST['info'];
64     $solusi = $_POST['solusi'];
65     $G001 = isset($_POST['G001']) ? 1 : 0;
66     $G002 = isset($_POST['G002']) ? 1 : 0;
67     $G003 = isset($_POST['G003']) ? 1 : 0;
68     $G004 = isset($_POST['G004']) ? 1 : 0;
69     $G005 = isset($_POST['G005']) ? 1 : 0;
70     $G006 = isset($_POST['G006']) ? 1 : 0;
71     $G007 = isset($_POST['G007']) ? 1 : 0;
72     $G008 = isset($_POST['G008']) ? 1 : 0;
73     $G009 = isset($_POST['G009']) ? 1 : 0;
74     $G010 = isset($_POST['G010']) ? 1 : 0;
75     $G011 = isset($_POST['G011']) ? 1 : 0;
76     $G012 = isset($_POST['G012']) ? 1 : 0;
77     $G013 = isset($_POST['G013']) ? 1 : 0;
78     $G014 = isset($_POST['G014']) ? 1 : 0;
79     $G015 = isset($_POST['G015']) ? 1 : 0;
80     $G016 = isset($_POST['G016']) ? 1 : 0;
81     $G017 = isset($_POST['G017']) ? 1 : 0;
82     $G018 = isset($_POST['G018']) ? 1 : 0;
83     $G019 = isset($_POST['G019']) ? 1 : 0;
84     $G020 = isset($_POST['G020']) ? 1 : 0;
85     $G021 = isset($_POST['G021']) ? 1 : 0;
```

Gambar 6.10 Coding Edit Data Penyakit (1)

```

86   $G022 = isset($_POST['G022']) ? 1 : 0;
87   $G023 = isset($_POST['G023']) ? 1 : 0;
88   $G024 = isset($_POST['G024']) ? 1 : 0;
89   $G025 = isset($_POST['G025']) ? 1 : 0;
90   $G026 = isset($_POST['G026']) ? 1 : 0;
91   $G027 = isset($_POST['G027']) ? 1 : 0;
92   $G028 = isset($_POST['G028']) ? 1 : 0;
93   $G029 = isset($_POST['G029']) ? 1 : 0;
94   $G030 = isset($_POST['G030']) ? 1 : 0;
95   $G031 = isset($_POST['G031']) ? 1 : 0;
96   $G032 = isset($_POST['G032']) ? 1 : 0;
97   $G033 = isset($_POST['G033']) ? 1 : 0;

```

Gambar 6.11 Coding Edit Data Penyakit (2)

Pada proses edit data penyakit, menggunakan *isset \$\_POST edit-gjl* untuk mengecek tombol *edit-gjl* ditekan atau tidak pada coding menu edit data penyakit.

```

<a href="data-penyakit.php" class="btn btn-danger">Kembali</a>
<button type="submit" name="edit-gjl" class="btn btn-success">Edit
Data</button>

```

Gambar 6.12 Coding Tombol *edit-gjl*

Jika tombol *edit-gjl* ditekan, maka akan menampilkan data yang berisi value dari variabel *\$penyakit, \$info, \$solusi, \$G001 - \$G033* berdasarkan id yang dipilih.

```

99   $sql = "UPDATE tb_penyakit SET penyakit='$penyakit',info='$info',solusi='$solusi' WHERE id=$id.";
100   $sql .= "UPDATE tb_risa SET G001=$G001,G002=$G002,G003=$G003,G004=$G004,G005=$G005,G006=$G006,G007=$G007,";
101   $sql .= "G008=$G008,G009=$G009,G010=$G010,G011=$G011,G012=$G012,G013=$G013,G014=$G014,G015=$G015,G016=$G016,";
102   $sql .= "G017=$G017,G018=$G018,G019=$G019,G020=$G020,G021=$G021,G022=$G022,G023=$G023,G024=$G024,G025=$G025,G026=$G026,";
103   $sql .= "G027=$G027,G028=$G028,G029=$G029,G030=$G030,G031=$G031,G032=$G032,G033=$G033 WHERE id=$id.";
104   if (mysql_multi_query($conn, $sql)) {
105       $message = "Data penyakit berhasil diedit";
106   } else {
107       $message = "Data penyakit gagal diedit" . mysql_error($conn);
108   }
109 }
110 }
111 }

```

Gambar 6.13 Coding Edit Data Penyakit (3)

Kemudian buat dua kueri *SQL* dengan *UPDATE* *tb\_penyakit* SET *penyakit=\$penyakit* dst dan *UPDATE* *tb\_rule* SET *\$G001=\$G033* dan seterusnya untuk mengedit data didalam tabel *tb\_penyakit* dan data didalam tabel *tb\_rule*. Lalu eksekusi kueri menggunakan *mysqli\_multi\_query*, jika berhasil maka data didalam tabel *tb\_penyakit* dan data didalam tabel *tb\_rule* akan diganti dengan data yang baru dan akan muncul pesan data penyakit berhasil diubah, dan jika tidak berhasil maka data didalam tabel *tb\_penyakit* dan data didalam tabel *tb\_rule* tidak akan berubah dan akan muncul pesan data penyakit tidak berhasil diubah.

### ***Coding* untuk cek penyakit**

```
1 <?php
2 $conn=mysqli_connect("localhost","root","","db_sp");
3 if(mysqli_connect_error()){
4     echo "Failed to Connect to Mysql : ".mysqli_connect_error();
5 }
6 ?>
```

Gambar 6.14 *Coding* koneksi.php

```

1 <?php
2 include 'koneksi.php';
3 if(isset($_POST['cek-pny']))){
4     $query='select id from tb_rule where ';
5     array_pop($_POST);
6     $rule_input=array();
7     foreach ($_POST as $kode) {
8         $query = $query . $kode . "=1 and ";
9         array_push($rule_input,$kode);
10    }
11    $query = $query . "1=1";
12
13    $data=mysqli_query($conn,$query);
14    $id='';

```

Gambar 6.15 Coding Cek Penyakit (1)

Membuat koneksi dengan database db\_sp menggunakan *include* dari file koneksi.php, lalu mengecek apakah tombol *cek-pny* sudah ditekan dengan menggunakan *isset(\$\_POST['cek-pny'])* dan mengambil data dari *form method post*, kemudian buat kueri dan inialisasi kueri untuk menyeleksi id dari tabel *tb\_rule*. Lalu hapus *array* indeks terakhir dengan *array\_pop(\$\_POST)* karena *array* indeks terakhir adalah tombol *cek-pny* kemudian simpan dalam variabel *\$input*. Lalu buat variabel *\$rule\_input* dengan isi *array* kosong yang nanti isinya adalah data yang diinputkan *user*, lakukan perulangan *\$\_POST* dan simpan setiap perulangan kedalam variabel *\$kode*. Kemudian gabungkan variabel *\$query* dengan variabel *\$kode* ditambah “=1 and 1” setelah perulangan,

kemudian tambahkan *value* kedalam *\$rule\_input* yang isinya *value* dari variabel *\$kode* hasil perulangan, lalu menggabungkan variabel *\$query* dengan "1=1", eksekusi *\$query* dengan menggunakan *mysqli\_query* dan disimpan dalam variabel *\$data*. Lalu buat variabel *\$id* dengan string kosong.

```
16     $db_rule=mysqli_query($conn,"select * from tb_rule");
17     $arr_rule = array();
18     while ($dt=mysqli_fetch_array($db_rule)) {
19         array_push($arr_rule,$dt);
20     }
```

Gambar 6.16 Coding Cek Penyakit (2)

Kemudian mengambil semua data rule dari tabel *tb\_rule* menggunakan *mysqli\_query* dan disimpan didalam variabel *\$db\_rule*, buat variabel *\$arr\_rule* yang isinya *array* kosong. Lakukan perulangan untuk mengambil setiap baris *\$db\_rule* kemudian simpan semu hasil perulangan didalam variabel *\$dt*, lalu *value* dari *\$dt* ditambahkan ke dalam *\$arr\_rule*.

```

23     $rule=array();
24     for ($i=0; $i <sizeof($arr_rule) ; $i++) {
25         $key=array_keys($arr_rule[$i]);
26         $value=$arr_rule[$i];
27         $sub_rule=array();
28         for($j=3;$j<(sizeof($key));$j+=2){
29             if($value[$key[$j]]==1)
30                 array_push($sub_rule, $key[$j]);
31         }
32         array_push($rule, $sub_rule);
33     }

```

Gambar 6.17 Coding Cek Penyakit (3)

Buat variabel *\$rule* yang isinya *array* kosong yang nanti isinya adalah semua data yang nilainya 1 dari variable *\$arr\_rule*, lalu dilakukan pengulangan untuk variabel *\$arr\_rule* dengan variabel *\$i* sebagai indeks untuk mengakses setiap elemen *array* dalam *\$arr\_rule* dengan setiap perulangan nilai *\$i* ditambah 1 selama nilai *\$i* kurang dari jumlah data dalam variabel *\$arr\_rule*. Dalam setiap perulangan, *\$key* di inialisasi dengan *array\_keys(\$arr\_rule[\$i])*, fungsi ini akan mengembalikan *array* yang berisi semua *key* dalam *\$arr\_rule* yang indeks nya sama dengan *\$i*. kemudian buat variabel *\$value* yang isinya sama dengan elemen array *\$arr\_rule* yang indeksnya *\$i*, lalu buat variabel *\$sub\_rule* yang isinya *array* kosong, lalu lakukan perulangan terhadap *\$key* dengan indeks variabel *\$j* yang dimulai dengan indeks 3 dengan pertambahan

indeks 2 karena elemen dengan indeks 3,5,7 dan seterusnya dalam *array \$key* memiliki nilai 1 atau 0 didalam tabel *tb\_rule*, perulangan akan terus berlangsung selama nilai variabel *\$j* kurang dari jumlah data dalam variabel *\$key*. pada setiap perulangan lakukan pemeriksaan if *value* dari *\$value[\$key[\$j]]==1*, jika benar maka dimasukan ke dalam array *\$sub\_rule* menggunakan *array\_push*. Setelah selesai, kemudian value dari *\$sub\_rule* dimasukan ke dalam *\$rule* dengan menggunakan *array\_push*.

```
35     $status=false;
36     for ($i=0; $i <sizeof($rule); $i++) {
37         $result=($rule_input==$rule[$i]);
38         if ($result) {
39             $status=true;
40         }
41     }
```

Gambar 6.18 Coding Cek Penyakit (4)

Buat variabel *\$status* dengan nilai *false*, lalu lakukan perulangan untuk setiap elemen dalam variabel *\$rule* dengan indeks variabel *\$i* dengan penambahan indeks 1, perulangan akan terus berlangsung selama nilai dari variabel *\$i* kurang dari jumlah data dalam variabel *\$rule*, pada setiap perulangan lakukan perbandingan nilai *\$rule\_input* yang berisi *value* yang diinputkan user dan *\$rule[\$i]*

dengan nilai yang sama dan hasilnya disimpan dalam variabel *\$result*, perbandingan ini menghasilkan nilai *true* jika kedua *array* memiliki elemen yang sama dan dalam urutan yang sama.

```
43     if($status==true){
44         while ($dt=mysqli_fetch_array($data)) {
45             $id=$dt['id'];
46         }
47         $cari_penyakit="select * from tb_penyakit where id=$id";
48         $db=mysqli_query($conn,$cari_penyakit);
49         while ($dt=mysqli_fetch_array($db)) {
50             $penyakit=$dt['penyakit'];
51             $info=$dt['info'];
52             $solusi=$dt['solusi'];
53             include 'hasil_diagnosa.php';
54         }
55     }else{
56         include 'eror.php';
57     }
58 }
59 ?>
```

Gambar 6.19 Coding Cek Penyakit (5)

Jika nilai *\$status* adalah *true*, maka lakukan perulangan untuk mengambil setiap baris hasil kueri variabel *\$data* dengan *mysqli\_fetch\_array* yang kemudian disimpan didalam variabel *\$dt*. Lalu value *\$dt* dimasukkan kedalam variabel *\$id* berdasarkan id nya, kemudian menyeleksi id dalam tabel *tb\_penyakit* lalu hasilnya disimpan didalam variable *\$cari\_penyakit*, kemudian eksekusi kueri *\$cari\_penyakit* dengan menggunakan *mysqli\_query* lalu hasilnya disimpan dalam variabel *\$db*. Lalu lakukan perulangan *\$db* dengan *mysqli\_fetch\_array* dan hasilnya disimpan

kedalam variabel *\$dt*, kemudian *key* dari variabel *\$dt* yang nilainya penyakit disimpan dalam variabel *\$penyakit*, lalu *key* dari variabel *\$dt* yang nilainya info disimpan dalam variabel *\$info*, dan *key* dari variabel *\$dt* yang nilainya solusi disimpan dalam variabel *\$solusi*. Setelah itu hasilnya ditampilkan dalam file *hasil\_diagnosa.php*. Jika variabel *\$status* tidak bernilai *true*, maka tampilkan hasilnya dari file *eror.php*.

### **Coding untuk ubah password User**

```
1 <?php
2 session_start();
3 $conn = mysqli_connect("localhost", "root", "", "db_sp");
4
5 if (isset($_POST['edit-user'])) {
6     $id = $_POST['id'];
7     $password = $_POST['password'];
8
9     $sql = "UPDATE anggota SET password='$password' WHERE id=$id";
10    if (mysqli_query($conn, $sql)) {
11        $message = "Password berhasil diubah";
12    } else {
13        $message = "Password gagal diubah" . mysqli_error($conn);
14    }
15 }
```

Gambar 6.20 Coding Ubah Password

Memulai sesi PHP dengan *session\_start()*, kemudian membuat koneksi dengan database *db\_sp* dengan *mysqli\_connect*. Lalu mengecek apakah tombol *edit-user* ditekan atau tidak menggunakan *isset(\$\_POST['edit-user'])*.

```

<div class="form-group row">
  <div class="col text-right">
    <a href="user.php" class="btn btn-danger">Kembali</a>
    <a href="aksi-user.php?del&id=<?= $data['id'] ?>" class="btn badge-warning">Hapus Akun</a>
    <button type="submit" name="edit-user" class="btn btn-success">Ubah Password </button>
  </div>
</div>

```

Gambar 6.21 Coding Tombol *edit-user*

Kemudian mengambil *value* *id*, *nama*, dan *password* dari *form method post* setelah tombol *edit-user* ditekan yang disimpan pada variabel *\$id*, *\$nama*, dan *\$password*. Lalu buat kueri *SQL* untuk mengubah data *password* didalam tabel anggota dengan *password='\$password'* dimana *id=\$id*. Kemudian eksekusi kueri tersebut, jika berhasil maka data *password* didalam tabel anggota akan diubah dan muncul informasi bahwa *password* berhasil diubah, jika gagal maka data *password* didalam tabel anggota tidak akan berubah dan muncul informasi bahwa *password* gagal diubah.

## Coding untuk hapus akun User

```
16 else if (isset($_GET['del'])) {
17     $id = $_GET['id'];
18
19     $sql = "delete from anggota where id=$id;";
20     if (mysqli_multi_query($conn, $sql)) {
21         $message = "Akun berhasil dihapus";
22         session_destroy();
23     } else {
24         $message = "Akun tidak terhapus" . mysqli_error($conn);
25     }
26 }
27 ?>
```

Gambar 6.20 Coding Hapus Akun User

Memulai sesi PHP dengan `session_start()`, lalu membuat koneksi dengan database yang memiliki nama `db_sp` dengan `mysqli_connect`, kemudian menggunakan `isset $_GET` untuk mengambil url yang berisi parameter `del` berdasarkan `id`. Setelah itu buat kueri `SQL` untuk menghapus semua data akun didalam tabel anggota dengan `id=$id`, lalu eksekusi kueri. Jika berhasil maka semua data akun dengan `id` sekian didalam tabel anggota akan dihapus dan muncul informasi bahwa akun berhasil dihapus kemudian lakukan `session_destroy()`; untuk menghapus semua sesi, jika gagal maka data akun didalam tabel anggota tidak terhapus dan muncul informasi bahwa akun tidak terhapus.

## Database db\_sp



Table	Action	Rows	Type	Collation	Size	Overhead
anggota	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tb_gejala	Browse Structure Search Insert Empty Drop	33	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tb_penakit	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tb_rule	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	53	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Gambar 7.1 Structure Database db\_sp

Database db\_sp adalah sebuah database yang digunakan sebagai wadah untuk menyimpan data-data yang digunakan didalam website sistem pakar dalam penelitian ini yang berisi beberapa tabel seperti, tabel anggota yang isinya data *user* dan admin, kemudian tabel tb\_gejala yang isinya data gejala-gejala, kemudian tabel tb\_penakit yang isinya data penyakit, dan tabel tb\_rule yang isinya rule-rule pengetahuan seorang pakar untuk mencocokkan antara data gejala dengan penyakit.



		id	nama	password	level
<input type="checkbox"/>	Edit Copy Delete	1	admin	admin	1
<input type="checkbox"/>	Edit Copy Delete	2	adi	555	0

Gambar 7.2 Tabel anggota

	id	kode	gejala
<input type="checkbox"/> Edit Copy Delete	1	G001	Diare
<input type="checkbox"/> Edit Copy Delete	2	G002	Nafas sesak/megap-megap
<input type="checkbox"/> Edit Copy Delete	3	G003	Nafas ngorok
<input type="checkbox"/> Edit Copy Delete	4	G004	Nafas cepat
<input type="checkbox"/> Edit Copy Delete	5	G005	Bersin-bersin
<input type="checkbox"/> Edit Copy Delete	6	G006	Batuk
<input type="checkbox"/> Edit Copy Delete	7	G007	Badan kurus
<input type="checkbox"/> Edit Copy Delete	8	G008	Bulu kusam dan berkerut
<input type="checkbox"/> Edit Copy Delete	9	G009	Nafsu makan berkurang
<input type="checkbox"/> Edit Copy Delete	10	G010	Produksi telur menurun
<input type="checkbox"/> Edit Copy Delete	11	G011	Kualitas telur jelek
<input type="checkbox"/> Edit Copy Delete	12	G012	Keliatan ngantuk dan bulu berdiri
<input type="checkbox"/> Edit Copy Delete	13	G013	Kedinginan
<input type="checkbox"/> Edit Copy Delete	14	G014	Tampak lesu
<input type="checkbox"/> Edit Copy Delete	15	G015	<b>Mencret kehijau-hijauan</b>
<input type="checkbox"/> Edit Copy Delete	16	G016	Mencret keputih-putihan
<input type="checkbox"/> Edit Copy Delete	17	G017	Mencret bercampur darah
<input type="checkbox"/> Edit Copy Delete	18	G018	Banyak minum
<input type="checkbox"/> Edit Copy Delete	19	G019	Muka pucat
<input type="checkbox"/> Edit Copy Delete	20	G020	Nampak membiru
<input type="checkbox"/> Edit Copy Delete	21	G021	Sempoyongan
<input type="checkbox"/> Edit Copy Delete	22	G022	Jengger membengkak merah
<input type="checkbox"/> Edit Copy Delete	23	G023	Kaki meradang/lumpuh

Gambar 7.3 Tabel tb\_gejala

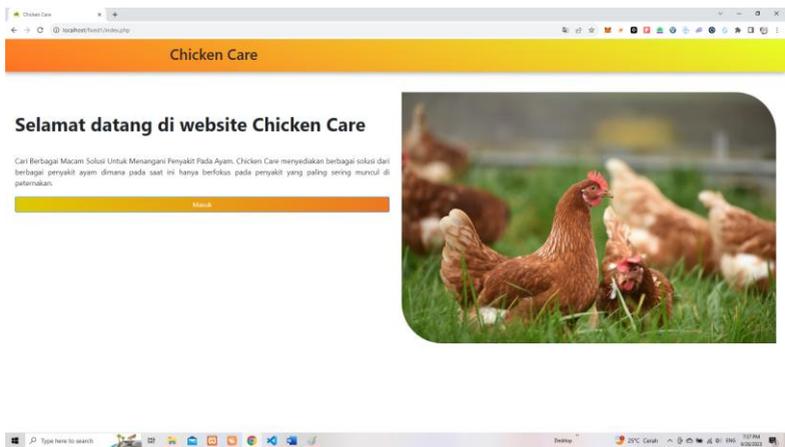
	id	penyakit	info	solusi
<input type="checkbox"/> Edit Copy Delete	1	EDS 76/Egg Drop Syndrome	Egg Drop Syndrome 76 merupakan penyakit infeksius	Pencegahan - melakukan vaksinasi EDS pada ayam
<input type="checkbox"/> Edit Copy Delete	2	Infectious Coryza (SNOT)	Infectious Coryza (SNOT) merupakan salah satu peny...	Pencegahan - melakukan vaksinasi kepada ayam
<input type="checkbox"/> Edit Copy Delete	3	ND (Newcastle Disease)	ND (Newcastle Disease) atau sering disebut tetelo...	Pencegahan - penanganan biosecurity dengan cara
<input type="checkbox"/> Edit Copy Delete	4	AI (Avian Influenza)	AI (Avian Influenza) merupakan penyakit virus yang...	Pencegahan - vaksinasi dengan Medivac ND-AI
<input type="checkbox"/> Edit Copy Delete	5	Infectious Bronchitis (IB)	Infectious Bronchitis (IB) merupakan penyakit vira...	Pencegahan - vaksinasi Medivac (B) H-120 atau Me...
<input type="checkbox"/> Edit Copy Delete	6	Koksidiosis	Koksidiosis atau sering disebut berak darah adalah...	Pencegahan - melakukan pemberian kapur atau sod...
<input type="checkbox"/> Edit Copy Delete	7	Infectious Bursal Disease (IBD)	Infectious Bursal Disease (IBD) atau disebut juga p...	Pencegahan - vaksinasi yang lepas seperti Mediv...
<input type="checkbox"/> Edit Copy Delete	8	Fowl Cholera	Fowl Cholera adalah penyakit menular yang disebabkan...	Pencegahan - santasi kandang dengan Medisep at...
<input type="checkbox"/> Edit Copy Delete	9	Terndiksi Penyakit Coryza/SNOT	Infectious Coryza (SNOT) merupakan salah satu peny...	Tindakan Pencegahan yang dapat dilakukan adalah ...

Gambar 7.4 Tabel tb\_penyakit

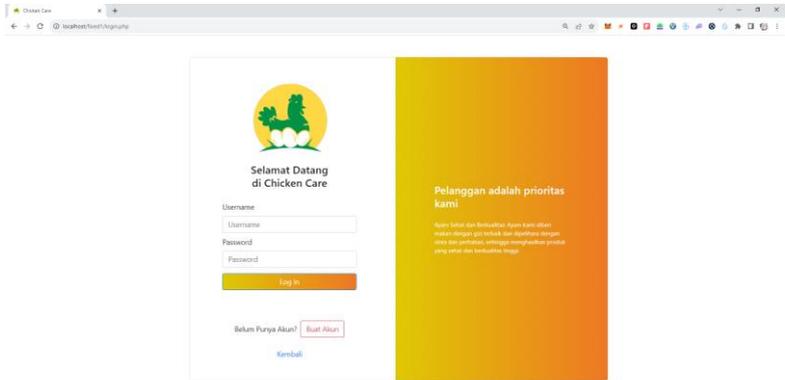
	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21
1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	1	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1
4	1	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1
5	1	0	1	0	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0
7	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0
8	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0
9	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Gambar 7.5 Tabel tb\_rule

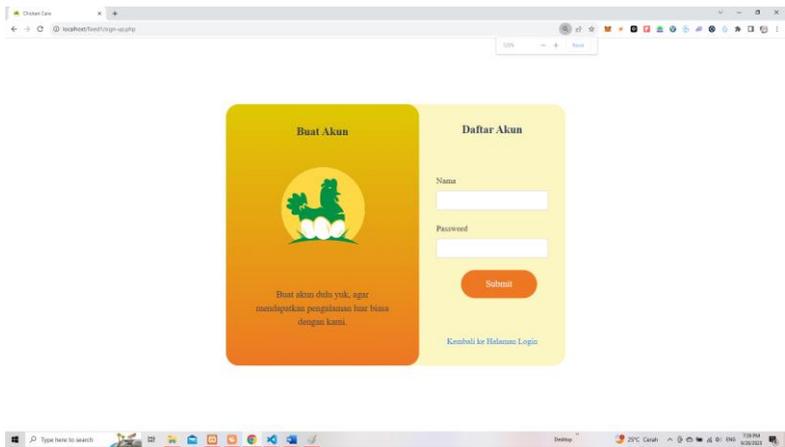
## b. Tampilan Hasil Coding



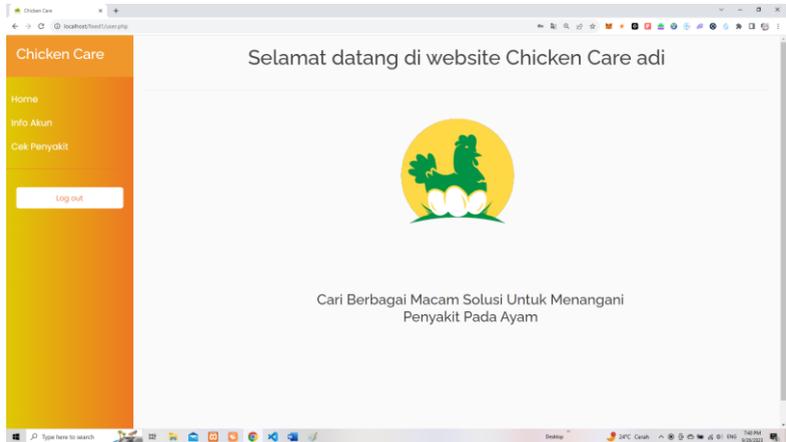
Gambar 8.1 Tampilan Halaman Awal Website



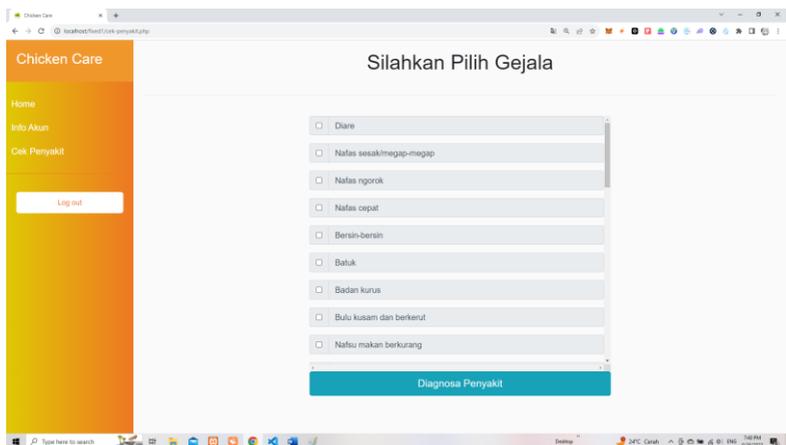
Gambar 8.2 Tampilan Halaman Login



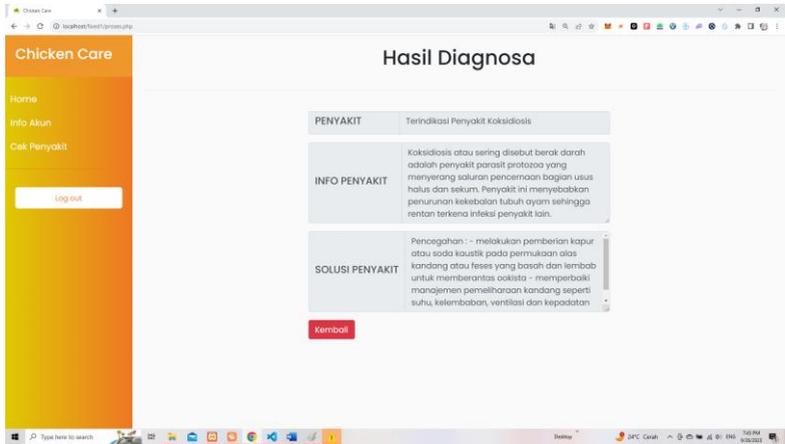
Gambar 8.3 Tampilan Halaman Buat Akun



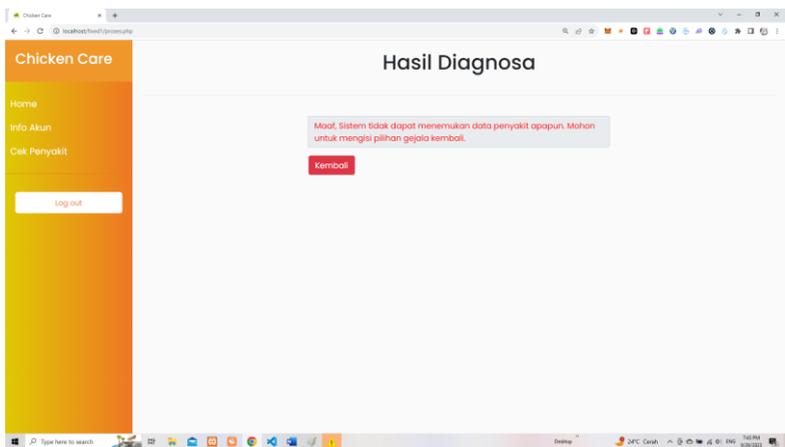
Gambar 8.4 Tampilan Halaman *User*



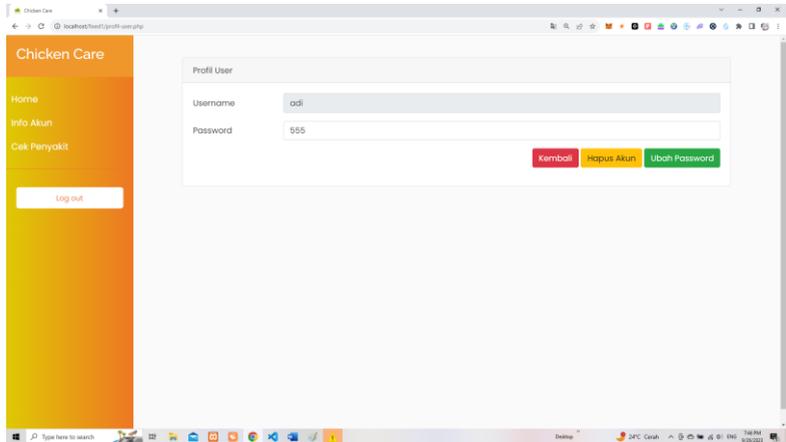
Gambar 8.5 Tampilan Halaman Cek Penyakit



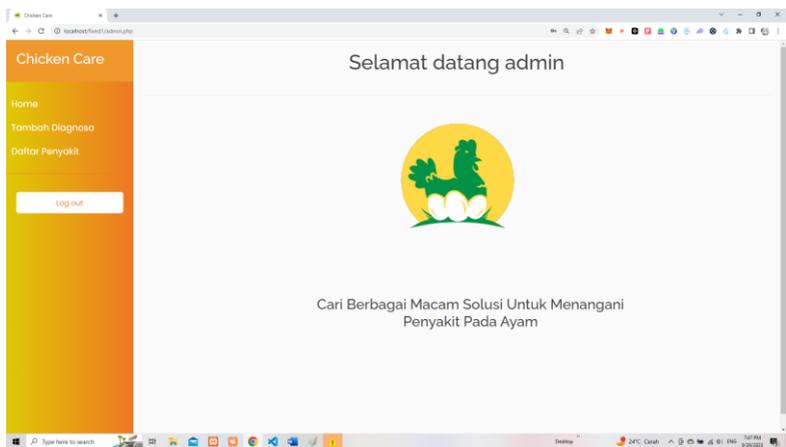
Gambar 8.6 Tampilan Halaman Hasil Diagnosa



Gambar 8.7 Tampilan Halaman Hasil Error

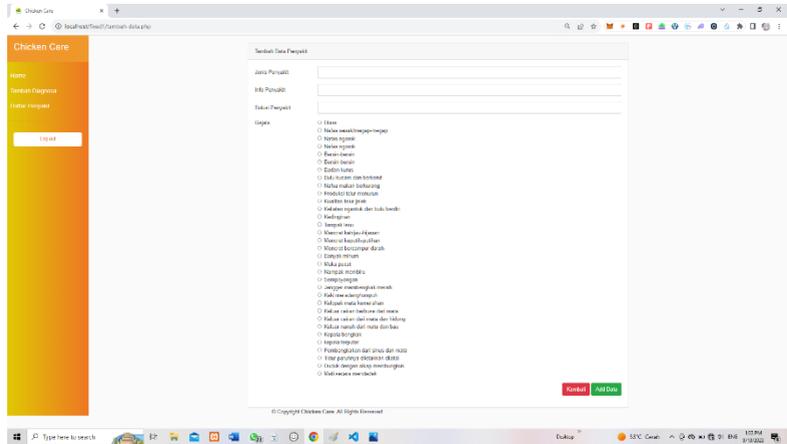


Gambar 8.8 Tampilan Halaman Info Akun *User*



Gambar 8.9 Tampilan Halaman Admin





Gambar 8.12 Tampilan Halaman Edit Data Penyakit

#### D. *Integration and Testing*

Pada proses *Integration and Testing* dilakukan penggabungan coding yang dilakukan pada proses implementation dan kemudian dilakukan pengujian apakah masih terdapat bug atau eror. Dalam penelitian ini menggunakan teknik *Black Box testing* dalam pengujiannya. Berikut hasil pengujian *Black Box* :

Pengujian reistrasi akun

Menu	Masukan	Contoh Data		Hasil Eksekusi
Registrasi	Data Kosong dan klik submit	Username	-	Tidak bisa registrasi, muncul notifikasi
		Password	-	

				harap isi kolom
	Data salah satu kosong dan klik submit	Username	adi	Tidak bisa registrasi, muncul notifikasi harap isi kolom
		Password	-	
	Data diisi semua dan klik submit	Username	adi	Berhasil login, muncul notifikasi bahwa sudah berhasil registrasi
		Password	555	
		Password	555	

Tabel 4.1 Tabel Pengujian Registrasi Akun

### Pengujian Login User

Menu	Masukan	Contoh Data		Hasil Eksekusi
Login User	Data Kosong dan klik login	Username	-	Tidak bisa login, muncul notifikasi harap isi kolom
		Password	-	
	Data salah satu kosong dan klik login	Username	adi	Tidak bisa login, muncul notifikasi harap isi kolom
		Password	-	
	Data diisi, salah satu data salah dan klik	Username	adi	Tidak bisa login, muncul notifikasi username atau
		Password	111	

	login			password salah
	Data diisi, data salah semua dan klik login	Username	awfawa	Tidak bisa login, muncul notifikasi username atau password salah
		Password	ghmfmr	
	Data diisi, data benar dan klik login	Username	adi	Berhasil login dan diarahkan ke halaman user
		Password	555	

Tabel 4.2 Tabel Pengujian Login *User*

### Pengujian Login Admin

Menu	Masukan	Contoh Data		Hasil Eksekusi
Login Admin	Data Kosong dan klik login	Username		Tidak bisa login, muncul notifikasi harap isi kolom
		Password		
	Data salah satu kosong dan klik login	Username	admin	Tidak bisa login, muncul notifikasi harap isi kolom
		Password		
	Data diisi, salah satu data salah dan klik login	Username	admin	Tidak bisa login, muncul notifikasi username atau password salah
		Password	111	
	Data diisi, data salah semua dan klik login	Username	hhuhuh	Tidak bisa login, muncul notifikasi username atau password salah
		Password	wjkrw	

	Data diisi, data benar dan klik login	Username	admin	Berhasil login dan diarahkan ke halaman user
		Password	admin	

Tabel 4.3 Tabel Pengujian Login Admin

### Pengujian Cek Penyakit

Menu	Masukan	Contoh Data		Hasil Eksekusi
Cek Penyakit	Gejala tidak dipilih dan klik diagnosa	Gejala	-	Muncul informasi untuk memilih dan melengkapi gejala yang benar
		Gejala	-	
	Gejala dipilih tapi salah dan klik diagnosa	Gejala	Nafsu makan berkurang	Muncul informasi untuk memilih dan melengkapi gejala yang benar
		Gejala	Badan kurus	
		Gejala	Mencret keputih-putihan	
		Gejala	Keluar cairan berbusa dari mata	
		Gejala	Muka pucat	
		Gejala	Mencret kehijau-hijauan	
	Gejala dipilih benar	Gejala	Nafsu makan berkurang	Muncul informasi terkait nama

dan klik diagnosa	Gejala	Badan kurus	penyakit, info penyakit, dan solusi penanganan dan pencegahannya
	Gejala	Bulu kusam dan berkerut	
	Gejala	Produksi telur menurun	
	Gejala	Muka pucat	
	Gejala	Mencret bercampur darah	
Gejala dipilih benar tapi kurang satu dan klik diagnosa	Gejala	Nafsu makan berkurang	Muncul informasi terkait nama penyakit, info penyakit, dan solusi penanganan dan pencegahannya
	Gejala	Badan kurus	
	Gejala	Bulu kusam dan berkerut	
	Gejala	Produksi telur menurun	
	Gejala	-	
Gejala dipilih benar tapi kurang dua dan klik diagnosa	Gejala	Nafsu makan berkurang	Muncul informasi terkait nama penyakit, info penyakit, dan solusi penanganan dan
	Gejala	Badan kurus	
	Gejala	-	
	Gejala	Produksi telur	

			menurun	pencegahannya
		Gejala	-	
		Gejala	Mencret bercampur darah	

Tabel 4.4 Tabel Pengujian Cek Penyakit

### Pengujian Tambah Data Penyakit

Menu	Masukan	Contoh Data		Hasil Eksekusi
Tambah data penyakit	Data Kosong kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
		Info Penyakit	-	
		Solusi Penyakit	-	
		Gejala	-	
	Hanya mengisi kolom jenis penyakit kemudian klik add data	Jenis Penyakit	A	Tidak bisa menambah data, muncul notifikasi harap isi kolom
		Info Penyakit	-	
		Solusi Penyakit	-	
		Gejala	-	
	Hanya mengisi kolom info penyakit kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
		Info Penyakit	A	
		Solusi Penyakit	-	
		Gejala	-	
Hanya mengisi kolom	Jenis Penyakit	-	Tidak bisa menambah data,	
	Info	-		

solusi penyakit kemudian klik add data	Penyakit		muncul notifikasi harap isi kolom
	Solusi Penyakit	A	
	Gejala	-	
Hanya memilih gejala kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	-	
	Solusi Penyakit	-	
	Gejala	Dipilih	
Hanya mengisi kolom jenis penyakit dan memilih gejala kemudian klik add data	Jenis Penyakit	A	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	-	
	Solusi Penyakit	-	
	Gejala	Dipilih	
Hanya mengisi kolom jenis penyakit dan kolom info penyakit kemudian klik add data	Jenis Penyakit	A	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	-	
	Gejala	-	
Hanya mengisi kolom	Jenis Penyakit	A	Tidak bisa menambah data,
	Info	-	

jenis penyakit dan solusi penyakit kemudian klik add data	Penyakit		muncul notifikasi harap isi kolom
	Solusi Penyakit	A	
	Gejala	-	
Hanya mengisi kolom info penyakit dan solusi penyakit kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	A	
	Gejala	-	
Hanya mengisi kolom info penyakit dan memilih gejala kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	-	
	Gejala	Dipilih	
Hanya mengisi kolom solusi penyakit dan memilih gejala kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	-	
	Solusi Penyakit	A	
	Gejala	Dipilih	
Mengisi	Jenis	A	Tidak bisa

kolom jenis penyakit, info penyakit, dan solusi penyakit kemudian klik add data	Penyakit		menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	A	
	Gejala	-	
Mengisi kolom jenis penyakit, info penyakit, dan memilih gejala kemudian klik add data	Jenis Penyakit	A	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	-	
	Gejala	Dipilih	
Mengisi kolom info penyakit, solusi penyakit, dan memilih gejala kemudian klik add data	Jenis Penyakit	-	Tidak bisa menambah data, muncul notifikasi harap isi kolom
	Info Penyakit	A	
	Solusi Penyakit	A	
	Gejala	Dipilih	
Mengisi kolom jenis penyakit,	Jenis Penyakit	A	Tidak bisa menambah data, muncul
	Info Penyakit	-	

	solusi penyakit, dan memilih gejala kemudian klik add data	Solusi Penyakit	A	notifikasi harap isi kolom
		Gejala	Dipilih	
	Mengisi semua data kemudian klik add data	Jenis Penyakit	A	Berhasil menambah data penyakit dan muncul informasi jika data berhasil diinput
		Info Penyakit	A	
		Solusi Penyakit	A	
		Gejala	Dipilih	

Tabel 4.5 Tabel Pengujian Tambah Data Penyakit

### Pengujian Edit Data Penyakit

Menu	Masukan	Contoh Data		Hasil Eksekusi
Edit data penyakit	Tidak mengubah apapun kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	-	
		Solusi Penyakit	-	
		Gejala	-	
	Hanya mengubah kolom jenis	Jenis Penyakit	B	Berhasil mengedit data,
		Info	-	

	penyakit kemudian klik edit data	Penyakit		muncul notifikasi data berhasil diedit
		Solusi Penyakit	-	
		Gejala	-	
	Hanya mengubah kolom info penyakit kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	B	
		Solusi Penyakit	-	
		Gejala	-	
	Hanya mengubah kolom solusi penyakit kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	-	
		Solusi Penyakit	B	
		Gejala	-	
	Hanya mengubah pilihan gejala kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	-	
		Solusi Penyakit	-	
		Gejala	Diubah	
	Hanya mengubah kolom jenis penyakit dan mengubah pilihan gejala	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit
Info Penyakit		-		
Solusi Penyakit		-		
Gejala		Diubah		

	kemudian klik edit data			
Hanya mengubah kolom jenis penyakit dan kolom info penyakit kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit	
	Info Penyakit	B		
	Solusi Penyakit	-		
	Gejala	-		
Hanya mengubah kolom jenis penyakit dan solusi penyakit kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit	
	Info Penyakit	-		
	Solusi Penyakit	B		
	Gejala	-		
Hanya mengubah kolom info penyakit dan solusi penyakit kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit	
	Info Penyakit	B		
	Solusi Penyakit	B		
	Gejala	-		
Hanya mengubah kolom info penyakit dan mengubah pilihan gejala kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit	
	Info Penyakit	B		
	Solusi Penyakit	-		
	Gejala	Diubah		

Hanya mengubah kolom solusi penyakit dan mengubah pilihan gejala kemudian klik edit data	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data berhasil diedit
	Info Penyakit	-	
	Solusi Penyakit	B	
	Gejala	Diubah	
Mengubah kolom jenis penyakit, info penyakit, dan solusi penyakit kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit
	Info Penyakit	B	
	Solusi Penyakit	B	
	Gejala	-	
Mengubah kolom jenis penyakit, info penyakit, dan mengubah pilihan gejala kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit
	Info Penyakit	B	
	Solusi Penyakit	-	
	Gejala	Diubah	
Mengubah kolom info penyakit, solusi penyakit, dan	Jenis Penyakit	-	Berhasil mengedit data, muncul notifikasi data
	Info Penyakit	B	
	Solusi Penyakit	B	

	mengubah pilihan gejala kemudian klik edit data	Gejala	Diubah	berhasil diedit
	Mengubah kolom jenis penyakit, solusi penyakit, dan mengubah pilihan gejala kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	-	
		Solusi Penyakit	B	
		Gejala	Diubah	
	Mengubah semua data kemudian klik edit data	Jenis Penyakit	B	Berhasil mengedit data, muncul notifikasi data berhasil diedit
		Info Penyakit	B	
		Solusi Penyakit	B	
		Gejala	Diubah	

Tabel 4.6 Tabel Pengujian Edit Data Penyakit

#### Pengujian Hapus Data Penyakit

Menu	Masukan	Contoh		Hasil Eksekusi
Hapus Data Penyakit	Delete	Tombol Delete	Di tekan	Berhasil menghapus data penyakit dan muncul informasi bahwa data

				penyakit berhasil dihapus
--	--	--	--	---------------------------

Tabel 4.7 Tabel Pengujian Hapus Data Penyakit

### Pengujian Ubah *Password*

Menu	Masukan	Contoh Data		Hasil Eksekusi
Ubah Password	Password dikosongi	Username	adi	Tidak bisa mengubah password dan muncul notifikasi untuk mengisi kolom password
		Password	-	
	Password diubah	Username	adi	Berhasil mengubah password dan muncul informasi password berhasil diubah
		Password	111	

Tabel 4.8 Tabel Pengujian Ubah *Password*

### Pengujian Hapus Akun User

Menu	Masukan	Contoh		Hasil Eksekusi
Hapus akun User	Hapus akun	Tombol hapus akun	Di tekan	Berhasil menghapus akun dan otomatis

				kembali ke halaman login
--	--	--	--	--------------------------

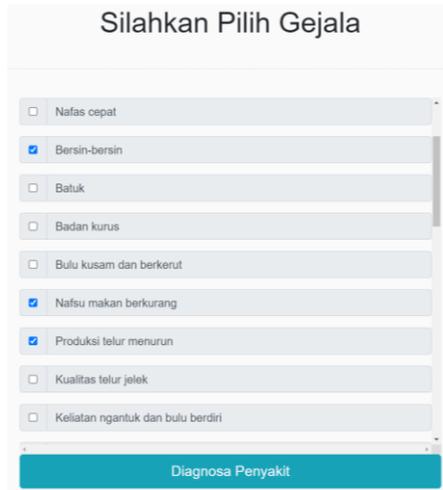
Tabel 4.9 Tabel Pengujian Hapus Akun User

### Pengujian Logout

Menu	Masukan	Contoh		Hasil Eksekusi
Logout User/Admin	Logout	Tombol logout	Di tekan	Berhasil logout dan kembali ke halaman awal

Tabel 4.10 Tabel Pengujian Logout

Uji coba cek penyakit dengan data yang didapat dari hasil wawancara :



Silahkan Pilih Gejala

- Nafas cepat
- Bersin-bersin
- Batuk
- Badan kurus
- Bulu kusam dan berkerut
- Nafsu makan berkurang
- Produksi telur menurun
- Kualitas telur jelek
- Keliatan ngantuk dan bulu berdiri

Diagnosa Penyakit

Gambar 9.1 Uji Pilih Gejala (1)

### Silahkan Pilih Gejala

- Keluar cairan berbusa dari mata
- Keluar cairan dari mata dan hidung
- Keluar nanah dari mata dan bau
- Kepala bengkok
- kepala terputar
- Pembengkakan dari sinus dan mata
- Tidur paruhnya diletakkan dilantai
- Duduk dengan sikap membungkuk
- Mati secara mendadak

Diagnosa Penyakit

Gambar 9.2 Uji Pilih Gejala (2)

### Hasil Diagnosa

<b>PENYAKIT</b>	Terindikasi Penyakit Coryza/SNOT
<b>INFO PENYAKIT</b>	Infectious Coryza (SNOT) merupakan salah satu penyakit bakterial yang menyerang sistem pernafasan ayam, terutama saluran pernafasan bagian atas. Korisa menyebabkan produksi telur turun hingga 10-40%, peningkatan ayam alkir, morbiditas hingga 20-50%.
<b>SOLUSI PENYAKIT</b>	Tindakan Pencegahan yang dapat dilakukan adalah : <ul style="list-style-type: none"> <li>- melakukan vaksinasi kepada ayam</li> <li>- pemberian vitamin dan elektrolit</li> <li>- melakukan desinfektan seperti Neo Antisept, Antisept, Medisept, dan Zalides</li> </ul>

Kembali

Gambar 9.3 Hasil Pengujian

Gejala yang dipilih menurut data dari hasil wawancara yaitu nafsu makan turun didalam pilihan gejala terdapat nafsu makan berkurang, kemudian gejala selanjutnya ada leleran seperti nanah dimata didalam pilihan gejala terdapat keluar nanah dari mata dan bau, kemudian gejala selanjutnya ada ayam cekreh, bersin didalam pilihan gejala terdapat bersin-bersin, kemudian gejala selanjutnya ada bengkak pada mata dan hidung didalam pilihan gejala terdapat pembengkakan dari sinus dan mata, dan gejala terakhir ada produksi turun didalam pilihan gejala terdapat produksi telur menurun. Setelah semua gejala dipilih kemudian saat di tekan tombol diagnosa penyakit maka muncul informasi seperti pada gambar 8.3, yaitu indikasi penyakit coryza dengan info dan solusi penyakitnya.

## BAB V

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Berdasarkan hasil penelitian yang dibahas pada bab sebelumnya, diperoleh beberapa poin kesimpulan :

1. Dalam website sistem pakar yang menggunakan metode *Forward Chaining* ini sudah dapat mendiagnosa penyakit dengan cukup baik, namun masih terdapat kekurangan dalam sistem pakar ini, yaitu belum bisa untuk mendeteksi penyakit lebih dari satu atau penyakit kompleks
2. Dalam pengujian *black box testing* sudah sesuai yang diharapkan yaitu fungsi tombol dan yang lainnya sudah bisa berjalan dengan baik.

#### B. Saran

Beberapa saran untuk penelitian selanjutnya yaitu mungkin untuk menambah beberapa fitur yang perlu ditambahkan.

## DAFTAR PUSTAKA

- Anhar. (2010). *Panduan Menguasai PHP dan MySQL Secara Otodidak*. Media Kita.
- A.S, R., & Salahudin, M. (2016). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (4th ed.). Informatika.
- Azura, A., & Wildian. (2018). Rancang Bangun Sistem Absensi Mahasiswa Menggunakan Sensor RFID dengan Database MySQL XAMPP dan Interface Visual Basic. *Jurnal Fisika Unand*, 7(2). <http://jfu.fmipa.unand.ac.id/index.php/jfu/article/view/350>
- Dewanti, R., & Sudiyono. (2017). PEMBERDAYAAN ANAK PANTI ASUHAN AN-NAHL DAN YAUMIHA DI KABUPATEN SRAGEN MELALUI KETRAMPILAN BETERNAK AYAM RAS PETELUR. *Jurnal SEMAR, ISSN 2302-3937*, 6(1 Nopember 2017), 67–77. <https://jurnal.uns.ac.id/jurnal-semar/article/view/20874>
- Endra, R. Y., & Antika, A. (2021). Sistem Pakar menggunakan Metode Forward Chaining untuk Diagnosa Penyakit Tanaman Padi berbasis Android. *Jurnal Informatika Universitas Pamulang* , 6(4), 811–817. <https://doi.org/10.32493/informatika.v6i4.14009>
- Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(1), 48. <https://doi.org/10.36448/expert.v11i1.2012>
- Febronius Bere, J., Dedy Irawan, J., & Ariwibisono, F. X. (2021). SISTEM PAKAR DIAGNOSIS PENYAKIT PADA AYAM MENGGUNAKAN METODE CERTAINTY FACTOR. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 5, Issue 1). <https://ejournal.itn.ac.id/index.php/jati/article/view/3251>
- Hidayat, A., & Yani, A. (2019). MEMBANGUN WEBSITE SMA PGRI GUNUNG RAYA RANAU MENGGUNAKAN PHP DAN MYSQL (Vol. 2, Issue 2). <https://journal.unmaha.ac.id/index.php/jtim/article/view/35>
- Kusbianto, D., Ardiansyah, R., & Hamadi, D. A. (2017). IMPLEMENTASI SISTEM PAKAR FORWARD CHAINING UNTUK IDENTIFIKASI DAN TINDAKAN PERAWATAN JERAWAT WAJAH. *Informatika Polinema*, 4(1), 71–80. <http://jip.polinema.ac.id/ojs3/index.php/jip/article/view/147/133>

- Mustofa, M. I. (2017). Rule Based Reasoning Untuk Monitoring Distribusi Bahan Bakar Minyak Secara Online dan Realtime menggunakan Radio Frequency Identification. *JURNAL SISTEM INFORMASI BISNIS*, 7(1), 39. <https://doi.org/10.21456/vol7iss1pp39-47>
- Pasalli, C. R., Poekoel, Vecky. C., & Najoan, X. (2016). Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining Berbasis Mobile. *E-Journal Teknik Informatika*, 7. <https://ejournal.unsrat.ac.id/index.php/informatika/article/view/12828>
- Putra, H. W., Yuhandri, & Nurcahyo, G. W. (2019). Sistem Pakar Diagnosis Penyakit Ginjal Dengan Metoda Forward Chaining. *Jurnal Sains Dan Informatika*, 5(1), 7–12. <https://doi.org/10.22216/jsi.v5i1.4081>
- Retno, F. D., Lestariningsih, C. L., Purwanto, B., & Hartono, S. (2015). *PENYAKIT-PENYAKIT PENTING PADA AYAM*. PT. Medion.
- Segara, A. (2019). Penerapan Pola Tata Letak (Layout Pattern) pada Wireframing Halaman Situs Web. *Jurnal Magenta*, 3(1), 452–464. <http://magenta.trisaktimultimedia.ac.id/magenta/index.php/magenta/article/view/45/35>
- Sommerville, I. (2011). *Software engineering*. Pearson.
- Sovia, R., & Febio, J. (2011). MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE. *Jurnal PROCESSOR*, 6.
- Tentua, M. N. (2009). SISTEM PAKAR DIAGNOSA PENYAKIT AYAM. *Dinamika Informatika*, 3(2), 95–110. [https://d1wqtxts1xzle7.cloudfront.net/34691528/SISTEM-PAKAR-DIAGNOSA-PENYAKIT-AYAM-libre.pdf?1410376767=&response-content-disposition=inline%3B+filename%3DSISTEM\\_PAKAR\\_DIAGNOSA\\_PENYAKIT\\_AYAM\\_Oleh.pdf&Expires=1679814783&Signature=D6qczWpDatjCLVeA60t215q4LPH89Y8P8KqTW3flz5qL4nCDc3XLeggYxwaPk1i14AtLCbSqxqyL3ndcrQnJQruEhAbuYkrrjTAoQHii7wDJUPzI1RrATfeC5fBswGyOOyO1RxjSzZQWUnNUwLnibyf7NqrG9jlpAa5fVQD1RoLVojYUU26smF86weX0uMRILtAN8Uzxxh~vr-0n7KT5YksHL4iT3RHhQ4K4EJCa41lsaimixvALhg8RNb7bmeuQ0UtXuY3sEtZcXFV77ZQZGLrRxujrSKvGZQdVH2aPAw17Pk9cChd0wDnQBzIUsgMNLTLTI89P1eZht6pFAt6~WQ\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/34691528/SISTEM-PAKAR-DIAGNOSA-PENYAKIT-AYAM-libre.pdf?1410376767=&response-content-disposition=inline%3B+filename%3DSISTEM_PAKAR_DIAGNOSA_PENYAKIT_AYAM_Oleh.pdf&Expires=1679814783&Signature=D6qczWpDatjCLVeA60t215q4LPH89Y8P8KqTW3flz5qL4nCDc3XLeggYxwaPk1i14AtLCbSqxqyL3ndcrQnJQruEhAbuYkrrjTAoQHii7wDJUPzI1RrATfeC5fBswGyOOyO1RxjSzZQWUnNUwLnibyf7NqrG9jlpAa5fVQD1RoLVojYUU26smF86weX0uMRILtAN8Uzxxh~vr-0n7KT5YksHL4iT3RHhQ4K4EJCa41lsaimixvALhg8RNb7bmeuQ0UtXuY3sEtZcXFV77ZQZGLrRxujrSKvGZQdVH2aPAw17Pk9cChd0wDnQBzIUsgMNLTLTI89P1eZht6pFAt6~WQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)

Wibowo, K. (2015). *ANALISA KONSEP OBJECT ORIENTED PROGRAMMING PADA BAHASA PEMROGRAMAN PHP* (Vol. 3, Issue DESEMBER).  
<https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/1662>