

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Analisa Kebutuhan *Software*

A. Tahapan Analisis

Berikut adalah spesifikasi kebutuhan (*system requirement*) yang penulis buat untuk aplikasi sistem pakar :

Halaman *User* :

- A.1. *User* dapat melakukan diagnosa
- A.2. *User* dapat melihat data pembuat aplikasi

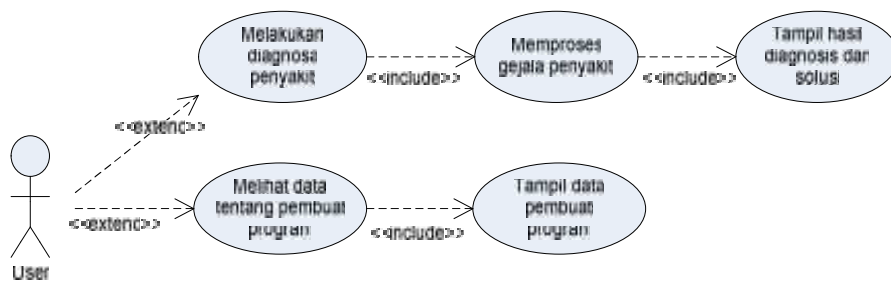
Halaman Admin :

- B.1. Admin dapat mengelola data gejala penyakit kulit pada kucing
- B.2. Admin dapat mengelola data penyakit kulit pada kucing
- B.3. Admin dapat mengelola data solusi penyakit kulit pada kucing

B. Use Case Diagram

Dari tahapan analisis, penulis membuat gambar *use case* sebagai berikut :

- 1. *Use case* diagram *User*

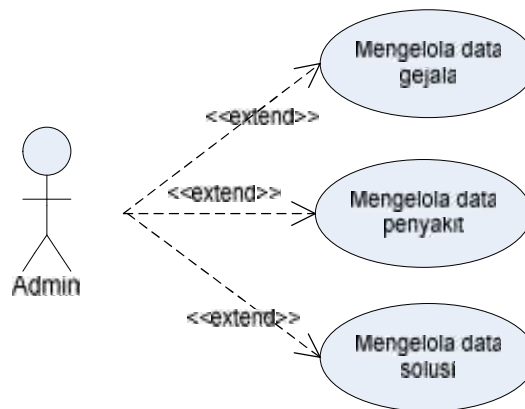


Gambar IV.1. Use Case Diagram User

Tabel IV.1. Deskripsi Use Case Diagram Penyakit Kulit Kucing halaman User

<i>Use Case Name</i>	Proses Diagnosa
<i>Requirement</i>	A1
<i>Goal</i>	User dapat mengakses segala menu yang ada di sistem pakar penyakit kulit pada kucing
<i>Pre-Conditions</i>	User mengetahui tentang aplikasi penyakit kulit pada kucing
<i>Post-Conditions</i>	User dapat mengetahui jenis penyakit dan solusi untuk menyembuhannya
<i>Failed and Conditions</i>	User tidak mendapatkan jenis penyakit ataupun solusi dari masalah pada diagnosa penyakit kucing yang di derita
<i>Primary Actors</i>	User
<i>Mainflow / Basic Path</i>	1. User menjawab pertanyaan berupa gejala pada penyakit kucing yang di derita 2. User mendapatkan informasi jenis penyakit kulit pada kucing 3. User mendapatkan solusi untuk jenis penyakit kulit pada kucing

2. Use case diagram Admin

**Gambar IV.2. Use Case Diagram Admin****Tabel IV.2. Deskripsi Use Case Diagram Data Gejala pada halaman Admin**

<i>Use Case Name</i>	Mengelola data gejala
<i>Requirement</i>	B1
<i>Goal</i>	Admin dapat mengelola data gejala
<i>Pre-Conditions</i>	Admin login ke dalam menu admin

<i>Post-Conditions</i>	Admin masuk ke dalam menu admin untuk mengakses data yang diperlukan
<i>Failed and Conditions</i>	Gagal <i>login</i> , gagal mengupdate data
<i>Primary Actors</i>	Admin
<i>Mainflow / Basic Path</i>	1. Admin menambah data gejala 2. Admin mengubah data gejala 3. Admin menghapus data gejala

Tabel IV.3. Deskripsi Use Case Diagram Data Penyakit pada halaman Admin

<i>Use Case Name</i>	Mengelola data penyakit
<i>Requirement</i>	B2
<i>Goal</i>	Admin dapat mengelola data penyakit
<i>Pre-Conditions</i>	Admin <i>login</i> ke dalam menu admin
<i>Post-Conditions</i>	Admin masuk ke dalam menu admin untuk mengakses data yang diperlukan
<i>Failed and Conditions</i>	Gagal <i>login</i> , gagal mengupdate data
<i>Primary Actors</i>	Admin
<i>Mainflow / Basic Path</i>	1. Admin menambah data penyakit 2. Admin mengubah data penyakit 3. Admin menghapus data penyakit

Tabel IV.4. Deskripsi Use Case Diagram Data Solusi pada halaman Admin

<i>Use Case Name</i>	Mengelola data solusi
<i>Requirement</i>	B3
<i>Goal</i>	Admin dapat mengelola data solusi
<i>Pre-Conditions</i>	Admin <i>login</i> ke dalam menu admin
<i>Post-Conditions</i>	Admin masuk ke dalam menu admin untuk mengakses data yang diperlukan
<i>Failed and Conditions</i>	Gagal <i>login</i> , gagal mengupdate data
<i>Primary Actors</i>	Admin
<i>Mainflow / Basic Path</i>	1. Admin menambah data solusi 2. Admin mengubah data solusi 3. Admin menghapus data solusi

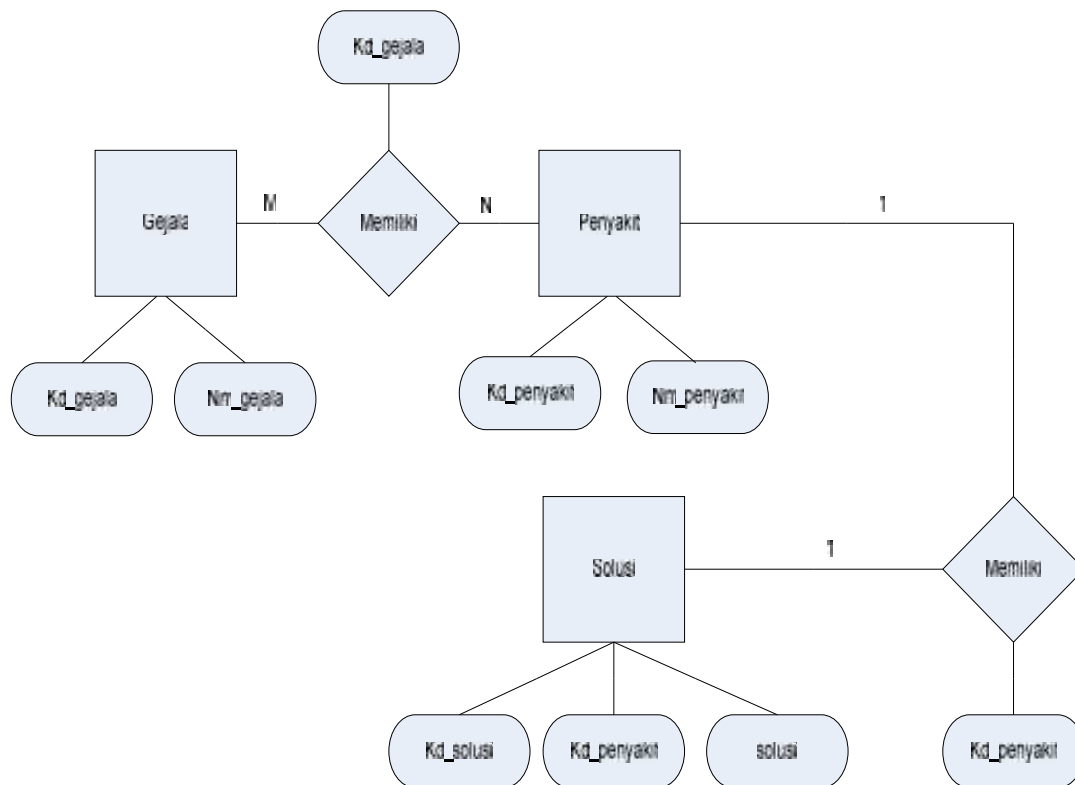
4.2. Desain

Pada tahapan ini akan dijelaskan tentang desain database, desain *software architecture* dan desain *user interface* dari sistem pakar ini.

4.2.1. Database

Penggambaran hubungan antar tabel yang dibuat serta relasi antar tabel pada sistem pakar ini menggunakan ERD, berikut adalah gambarnya :

1. Entity Relationship Diagram



Gambar IV.3. Entity Relationship Diagram

2. Spesifikasi File

a. Spesifikasi file Tabel Gejala

Nama File : File Gejala
 Akronim : Tabel Gejala
 File : Master
 Akses File : Random Access File

Primary File : kd_gejala

Panjang Record : 103

Tabel IV.5. Spesifikasi File Tabel Gejala

No.	Elemen Data	Nama Field	Type	Size	Keterangan
1	Kode Gejala	kd_gejala	Varchar	3	Primary Key
2	Nama Gejala	nm_gejala	Varchar	100	

b. Spesifikasi file Tabel Penyakit

Nama File : File Penyakit

Akronim : Tabel Penyakit

Type File : Master

Akses File : Random Access File

Primary Key : kd_penyakit

Panjang Record : 106

Tabel IV.6. Spesifikasi File Tabel Penyakit

No.	Elemen Data	Nama Field	Type	Size	Keterangan
1	Kode Penyakit	kd_penyakit	Varchar	3	Primary Key
2	Kode Gejala	kd_gejala	Varchar	3	Foreign Key
3	Nama Penyakit	nm_penyakit	Varchar	100	

c. Spesifikasi File Tabel Solusi

Nama File : File Solusi

Akronim : Tabel Solusi

Type File : Master

Akses File : Random Access File

Primary Key : kd_solusi

Panjang Record : 10

Tabel IV.7. Spesifikasi *File* Tabel Solusi

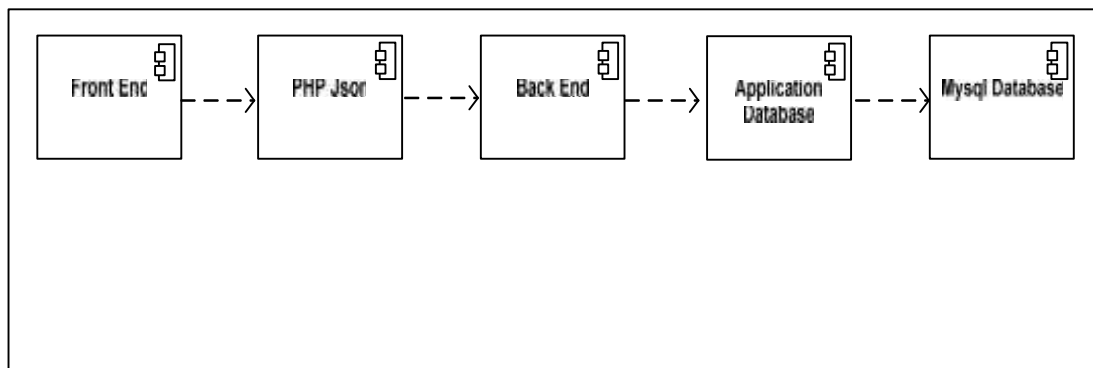
No.	Elemen Data	Nama <i>Field</i>	Type	Size	Keterangan
1	Kode Solusi	kd_solusi	Varchar	3	<i>Primary Key</i>
2	Kode Penyakit	kd_penyakit	Varchar	3	<i>Foreign Key</i>
3	Solusi	solusi	Text		

4.2.2. *Software Architecture*

Desain *software architecture* pada rancangan sistem dan program usulan ini dibuat dengan menggambarkan sebuah *component diagram* dan *deployment diagram*.

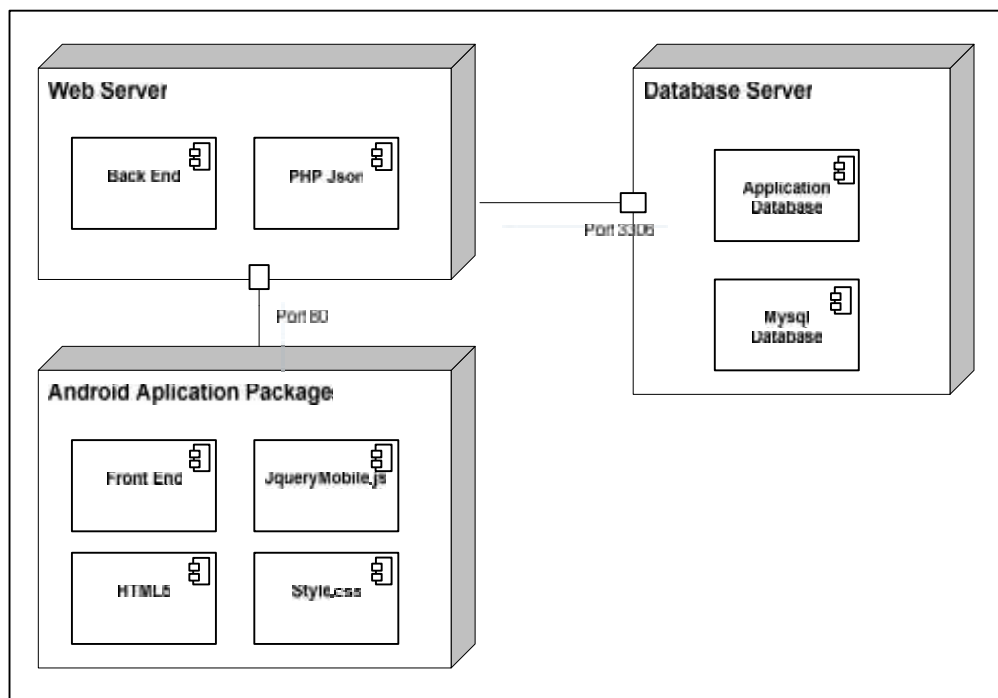
A. *Component Diagram*

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan antar kumpulan komponen di dalam sebuah sistem. *Component diagram* ini digunakan untuk memodelkan hubungan saling ketergantungan antara *source code* perangkat lunak (*software*), aplikasi *database* dan antarmuka (*interface*).

Gambar IV.4. *Component Diagram* Penyakit Kulit Pada Kucing

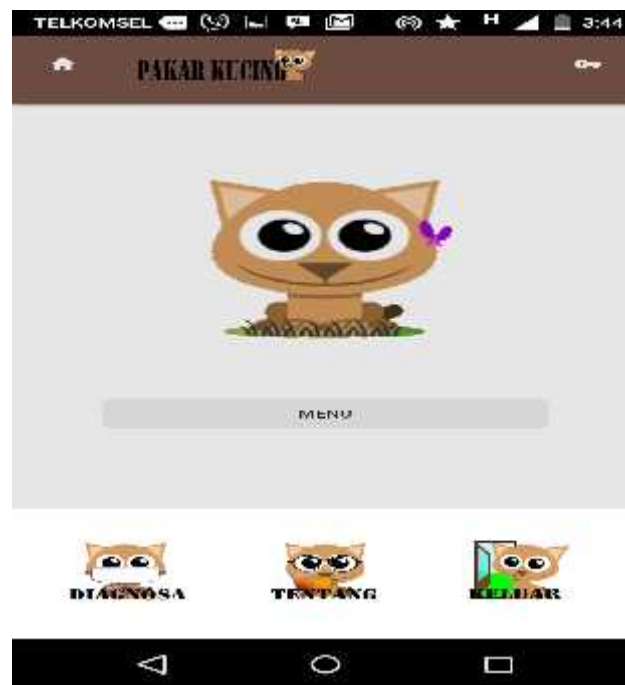
B. *Deployment Diagram*

Deployment diagram menggambarkan detail bagaimana komponen di *deployment* dalam infrastruktur sistem, menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.



. Gambar IV.5. *Deployment Diagram Penyakit Kulit Pada Kucing*

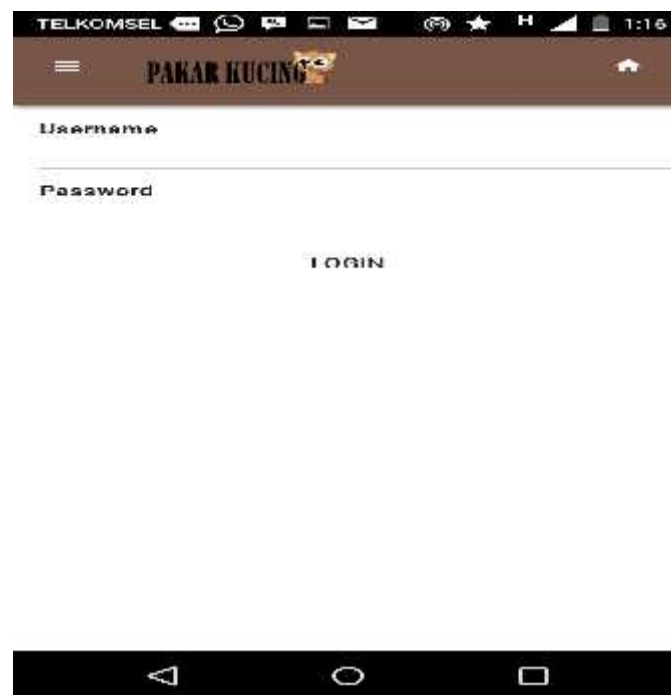
4.2.3. User Interface



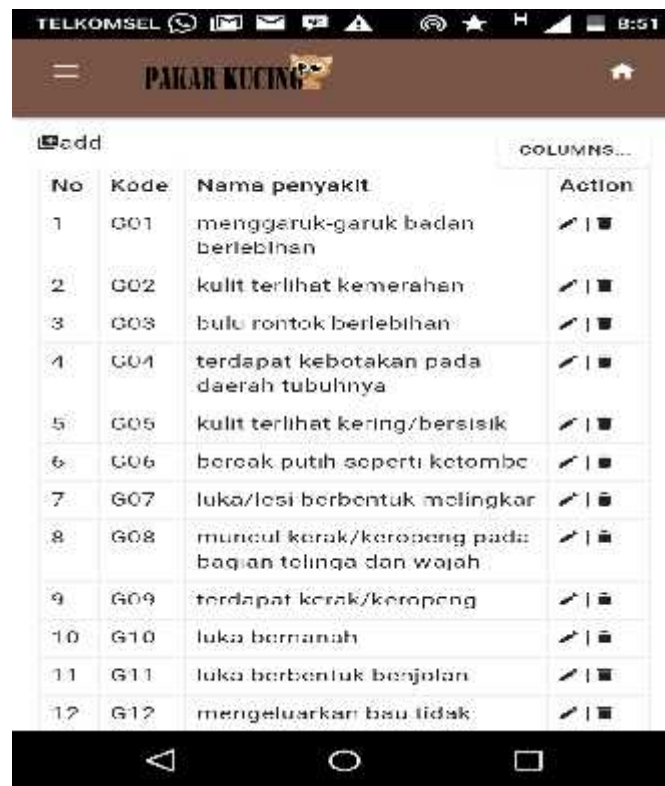
Gambar IV.6. Tampilan pada menu *User*



Gambar IV.7. Tampilan Diagnosa

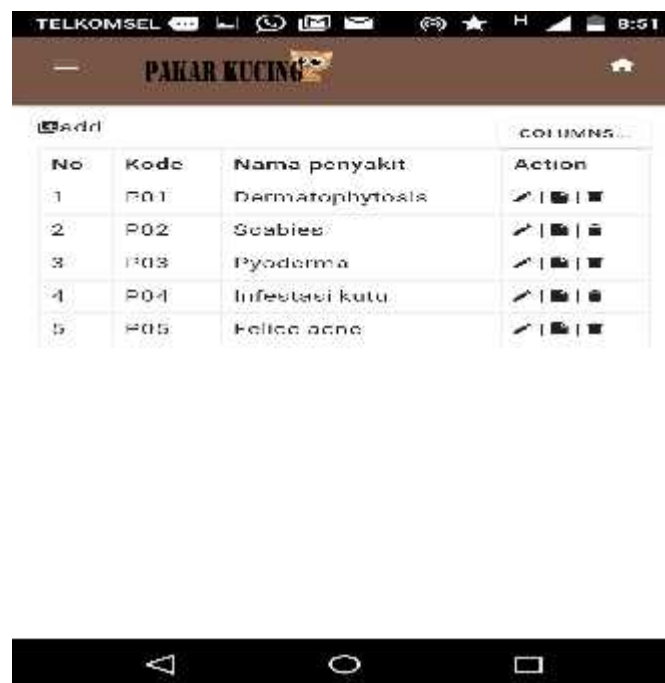


Gambar IV.8. Tampilan Login Admin



No	Kode	Nama penyakit	Action
1	G01	menggaruk-garuk badan berlebihan	✍️ 🗑️
2	G02	kulit terlihat kemerahan	✍️ 🗑️
3	G03	bulu rontok berlebihan	✍️ 🗑️
4	G04	terdapat kebotakan pada daerah tubuhnya	✍️ 🗑️
5	G05	kulit terlihat kering/bersisik	✍️ 🗑️
6	G06	bercak putih seperti ketombe	✍️ 🗑️
7	G07	luka/lesi berbentuk melingkar	✍️ 🗑️
8	G08	muncul kerak/keropeng pada bagian telinga dan wajah	✍️ 🗑️
9	G09	terdapat kerak/keropeng	✍️ 🗑️
10	G10	luka bernanah	✍️ 🗑️
11	G11	luka berbentuk benjolan	✍️ 🗑️
12	G12	mengeluarkan bau tidak	✍️ 🗑️

Gambar IV.9. Tampilan Data Gejala



No	Kode	Nama penyakit	Action
1	P01	Dermatophytosis	✍️ 🗑️ 📄
2	P02	Scabies	✍️ 🗑️ 📄
3	P03	Pyoderma	✍️ 🗑️ 📄
4	P04	Infestasi kutu	✍️ 🗑️ 📄
5	P05	Felice acne	✍️ 🗑️ 📄

Gambar IV.10. Tampilan Data Penyakit

No	Kode solusi	Kode penyakit	Action
1	AA	P01	[edit] [delete]
2	BB	P02	[edit] [delete]
3	CC	P03	[edit] [delete]
4	DD	P04	[edit] [delete]
5	EE	P05	[edit] [delete]

Gambar IV.11. Tampilan Data Solusi

4.3. Code Generation

Berikut adalah *code generation* yang ada pada sistem pakar penyakit kulit pada kucing :

1. Front End Diagnosa

```
//=====
// VARIABEL GLOBAL
//=====
var url = "http://192.168.43.162/"; //localhost
var gejala_list = [];
var diagnose_proses = false;

//=====
// GET DATA FUNCTION RETURN DATA
//=====
function getgejala()
{
    $.getJSON(url + 'service/mediapakar.php', {module:
"gejala_list"}, function(res) {
        gejala_list = res.items;
        $.each(gejala_list, function(i, value) {
```

```

        gejala_list[i].Y = gejala_list[i].N = '';
    });
});
}

function getdiagnoseprocess()
{
    window.history.back();
    //$('#[data-role=dialog]').dialog("close");
    var gejala_choice = [];
    for (i = 0; i < gejala_list.length; i++) {
        if (gejala_list[i].Y == 1)
            gejala_choice.push(gejala_list[i].kd_gejala);
    }
    gejala_choice = JSON.stringify(gejala_choice);
    $.ajax({
        url: url + 'service/mediapakar.php',
        data: { module: "diagnose_process", gejala_choices:
gejala_choice },
        traditional: true,
        success: function(res) {
            result = JSON.parse(res).item;
            opendiv("penyakitrowpage");
            $('#penyakitrow').append('<tr>');
            $('#penyakitrow').append('<td><b>nama
penyakit</b></td>');
            $('#penyakitrow').append('<td>' +
result.nm_penyakit + '</td>');
            $('#penyakitrow').append('</tr>');
            $('#penyakitrow').append('<tr>');
            $('#penyakitrow').append('<td
colspan="2"><center><img src="" + result.img_src + ""
width="300px" height="300px"></center></td>');
            $('#penyakitrow').append('</tr>');
            $('#penyakitrow').append('<tr>');
            $('#penyakitrow').append('<td
colspan="2"><b>gejala dipilih</b></td>');
            $('#penyakitrow').append('</tr>');
            for (i = 0; i < result.gejala_detail.length;
i++) {
                $('#penyakitrow').append('<tr>');
                $('#penyakitrow').append('<td
colspan="2">' + result.gejala_detail[i] + '</td>');
                $('#penyakitrow').append('</tr>');
            }
            $('#penyakitrow').append('<tr>');
            $('#penyakitrow').append('<td
colspan="2"><b>solusi</b></td>');
            $('#penyakitrow').append('</tr>');
            $('#penyakitrow').append('<tr>');
            $('#penyakitrow').append('<td colspan="2">' +
result.solusi + '</td>');

```

```

        $('#penyakitrow').append('</tr>');
    }
});
$('[data-role=dialog]').dialog("close");
}

//=====
// COMPARE DATA FUNCTION RETURN
//=====

//=====
// PROCES DATA MANIPULATION
//=====
function diagnose(index, y, n)
{
    // count
    document.getElementById("diagnose_count").innerHTML =
gejala_list.length;

    if ($.isNumeric(index)) {
        index = parseInt(index);

        // btn back
        document.getElementById("diagnose_back").innerHTML =
'';
        if (index != 0)
document.getElementById("diagnose_back").innerHTML = index - 1;

        // show
        document.getElementById("diagnose_next").innerHTML =
index + 1;

        document.getElementById("diagnose_nm_gejala").innerHTML =
gejala_list[index].nm_gejala;

        // btn start or end proses

        document.getElementById("diagnose_start").style.display =
document.getElementById("diagnose_end").style.display = 'none';
        if (index + 1 == gejala_list.length) {

            document.getElementById("diagnose_end").style.display = '';
        } else {

            document.getElementById("diagnose_start").style.display =
'';
        }

        // choice
        gejalachoice(index - 1, y, n);
    }
}

```

```

}

function gejalachoice(index, y, n)
{
    gejala_list[index].Y = y;
    gejala_list[index].N = n;
}

```

2. *Back End Diagnosa*

```

<?php
include "connection.php";
$module = NULL;
if (!empty($_GET['module'])) $module = $_GET['module'];
// MODULE PROCES DATA MANIPULATION
if ($module == 'diagnose_process') {
    try {
        $gejala_choices = NULL;
        if (!empty($_GET['gejala_choices'])) $gejala_choices =
$_GET['gejala_choices'];
        if (empty($gejala_choices)) {
            echo '{"error":{"text":"module not found !"}}';
            exit();
        }
        $gejala_choices = json_decode($gejala_choices, true);
        $sql = 'SELECT kd_penyakit, COUNT(*) as count FROM
relasi GROUP BY kd_penyakit';
        $stmt = $dbh->query($sql);
        $data = $stmt->fetchAll(PDO::FETCH_OBJ);
        foreach ($data as $k => $v) {
            $sql = 'SELECT kd_gejala FROM relasi WHERE
kd_penyakit="' . $v->kd_penyakit . '"';
            $stmt = $dbh->query($sql);
            $gejala_data = $stmt->fetchAll(PDO::FETCH_OBJ);
            $data[$k]->count = (int) $data[$k]->count;
            foreach ($gejala_data as $kk => $vv) {
                if (in_array($vv->kd_gejala,
$gejala_choices)) $data[$k]->count -= 1;
            }
        }
        $kd_penyakit = penyakit_count_min($data);
        $penyakit_diagnose = penyakit_diagnose($kd_penyakit,
$gejala_choices);
        echo '{"item":' . json_encode($penyakit_diagnose) . '}';
    } catch (PDOException $e) {
        echo '{"error":{"text":"' . $e->getMessage() . '}}';
    }
} else {
    echo '{"error":{"text":"module not found !"}}';
}
}

```

```
//=====
// FUNCTION PROCES DATA MANIPULATION
//=====
function penyakit_count_min($data)
{
    $count = array_map(function($res) {
        return $res->count;
    }, $data);
    $min = min($count);
    foreach ($data as $k => $v) {
        if ($v->count == $min) return $v->kd_penyakit;
    }
}

function penyakit_diagnose($kd_penyakit = NULL, $gejala_choices =
array())
{
    include "connection.php";
    $sql = 'SELECT * FROM penyakit WHERE kd_penyakit="' .
$kd_penyakit . '"';
    $stmt = $dbh->query($sql);
    $penyakit_diagnose = $stmt->fetchObject();
    foreach ($gejala_choices as $key => $val) {
        $gejala_row = gejala_row($val);
        $penyakit_diagnose->gejala_detail[$key] = $gejala_row-
>nm_gejala;
    }

    return $penyakit_diagnose;
}

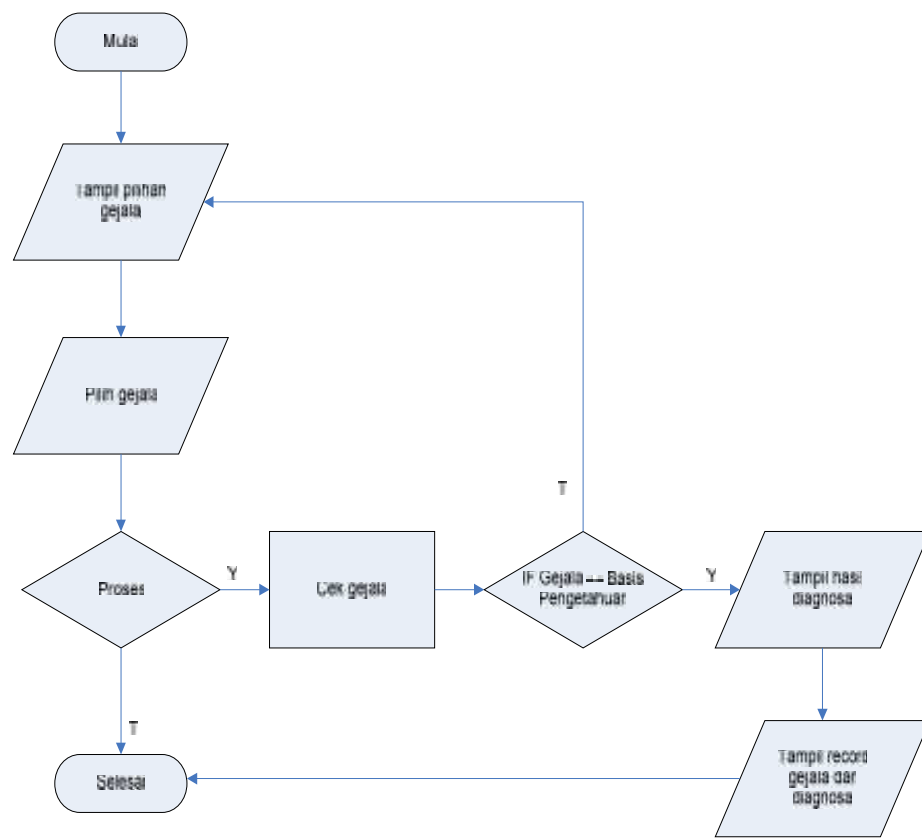
function gejala_row($kd_gejala = NULL)
{
    include "connection.php";
    $sql = 'SELECT * FROM gejala WHERE kd_gejala="' .
$kd_gejala . '"';
    $stmt = $dbh->query($sql);

    return $stmt->fetchObject();
}

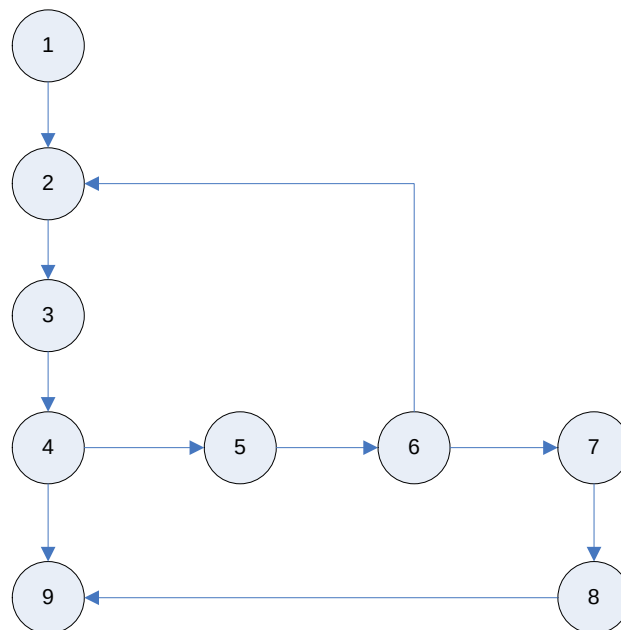
```

4.4. Testing

Pada tahap ini pengujian dilakukan dengan menggunakan metode pengujian-pengujian akan dilakukan terhadap form diagnosis dan berikut adalah *flowchart* dan *flowgraph form* penelusuran.



Gambar IV.12. Flowchart Menu Penelusuran



Gambar IV.13. Flowgraph Form Penelusuran

Sehingga kompleksitas siklomatisnya :

$$V(G) = E - N + 2$$

Dimana :

E = Jumlah edge grafik alir yang ditandakan dengan gambar panah.

N = Simpul grafik alir yang ditandakan dengan gambar lingkaran.

Sehingga kompleksitas siklomatisnya adalah :

$$V(G) = 10 - 9 + 2 = 3$$

Basis set yang dihasilkan dari jalur independent secara linier adalah jalur sebagai berikut :

1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

1 – 2 – 3 – 4 – 9

1 – 2 – 3 – 4 – 5 – 6 – 2

Ketika aplikasi dijalankan maka terlihat bahwa salah satu baris set yang dihasilkan adalah 1 – 2 – 3 – 4 – 9 dann terlihat bahwa simpul telah dieksekusi satu kali. Berdasarkan pengamatan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.

4.5. Support

Menjelaskan tentang perangkat pendukung pembuatan sistem pakar ini.

4.5.1. Spesifikasi *Hardware* dan *Software*

Pada tahapan ini perangkat lunak keras (*hardware*) dan perangkat lunak (*software*) yang digunakan untuk mendukung pada sistem pakar ini adalah sebagai berikut:

Tabel IV.8. Spesifikasi *Hardware* dan *Software*

Kebutuhan	Keterangan
Sistem Operasi	Android Kitkat ver 4.4.1 dan seterusnya
<i>Processor</i>	1.2 GHz quad core
RAM	1 GB
<i>Storage</i>	2 GB
<i>Screen</i>	4 inch
<i>Keyboard</i>	<i>Touch Screen</i>
<i>Software</i>	<i>Phonegap</i>