

BAB IV

IMPLEMENTASI SISTEM

4.1 Implementasi Basis Data

Software pengolah *database* yang digunakan dalam implementasi *database* sistem pakar untuk mendiagnosis penyakit pada ternak babi dengan metode *forward chaining* yaitu menggunakan *MySQL*. Berikut adalah tabel-tabel yang dibangun menjadi *database* pada sistem adalah:

4.1.1 Implementasi Tabel *Admin*

Tabel *admin* digunakan untuk merekam data *admin*. Tabel *admin* memiliki 3 *field* dengan *primary key*-nya adalah *id_admin*. Tabel *admin* dapat dilihat pada gambar di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	Tindakan
1	id_admin	int(6)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	username	varchar(25)		Tidak	Tidak ada		Ubah Hapus Lainnya
3	password	varchar(50)		Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar 4.1 Implementasi Tabel *Admin*

4.1.2 Implementasi Tabel Pakar

Tabel pakar digunakan untuk merekam data pakar. Tabel pakar memiliki 8 *field* dengan *primary key*-nya adalah *id_pakar*. Tabel pakar dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	Tindakan
1	id_pakar	int(6)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(25)		Tidak	Tidak ada		Ubah Hapus Lainnya
3	username	varchar(25)		Tidak	Tidak ada		Ubah Hapus Lainnya
4	password	varchar(50)		Tidak	Tidak ada		Ubah Hapus Lainnya
5	tempat_kerja	varchar(100)		Tidak	Tidak ada		Ubah Hapus Lainnya
6	lamatan	varchar(100)		Tidak	Tidak ada		Ubah Hapus Lainnya
7	gambar	varchar(100)		Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar 4.2 Implementasi Tabel Pakar

4.1.3 Implementasi Tabel Peternak

Tabel peternak digunakan untuk merekam data peternak. Tabel peternak memiliki 6 *field* dengan *primary key*-nya adalah *id_peternak*. Tabel peternak dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_peternak	int(6)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(25)			Tidak	Tidak ada		Ubah Hapus Lainnya
3	username	varchar(25)			Tidak	Tidak ada		Ubah Hapus Lainnya
4	password	varchar(50)			Tidak	Tidak ada		Ubah Hapus Lainnya
5	jk	varchar(10)			Tidak	Tidak ada		Ubah Hapus Lainnya
6	alamat	text			Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar 4.3 Implementasi Tabel Peternak

4.1.4 Implementasi Tabel Penyakit

Tabel penyakit digunakan untuk merekam data penyakit. Tabel penyakit memiliki 5 *field* dengan *primary key*-nya adalah *kode_ penyakit*. Tabel penyakit dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	kode_penyakit	varchar(6)			Tidak	Tidak ada		Ubah Hapus Lainnya
2	nm_penyakit	varchar(100)			Tidak	Tidak ada		Ubah Hapus Lainnya
3	definisi	text			Tidak	Tidak ada		Ubah Hapus Lainnya
4	penyebab	text			Tidak	Tidak ada		Ubah Hapus Lainnya
5	solusi	text			Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar 4.4 Implementasi Tabel Penyakit

4.1.5 Implementasi Tabel Gejala

Tabel gejala digunakan untuk merekam data gejala. Tabel gejala memiliki 2 *field* dengan *primary key*-nya adalah *kode_ gejala*. Tabel gejala dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	kode_gejala	varchar(6)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Lainnya
2	nm_gejala	text			Tidak	Tidak ada		Ubah Hapus Kunci Utama Lainnya

Gambar 4.5 Implementasi Tabel Gejala

4.1.6 Implementasi Tabel *Rule*

Tabel rule digunakan untuk merekam data *rule*. Tabel rule memiliki 4 *field* dengan *primary key*-nya adalah *id_rule*. Tabel *rule* dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	Tindakan
1	<i>id_rule</i>	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah • Hapus • Lainnya
2	<i>kode_gejala</i>	varchar(6)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
3	<i>ya</i>	varchar(6)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
4	<i>tidak</i>	varchar(6)		Tidak	Tidak ada		Ubah • Hapus • Lainnya

Gambar 4.6 Implementasi Tabel *Rule*

4.1.7 Implementasi Tabel Konsultasi

Tabel konsultasi digunakan untuk merekam data konsultasi. Tabel konsultasi memiliki relasi dengan tabel peternak dan tabel penyakit. Tabel konsultasi memiliki 4 *field* dengan *primary key*-nya *id_konsul*. Tabel konsultasi dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	Tindakan
1	<i>id_konsul</i>	int(6)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah • Hapus • Lainnya
2	<i>id_peternak</i>	int(11)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
3	<i>kode_penyakit</i>	varchar(100)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
4	<i>tgl_diagnosa</i>	date		Tidak	Tidak ada		Ubah • Hapus • Lainnya

Gambar 4.7 Implementasi Tabel Konsultasi

4.1.8 Implementasi Tabel Pesan

Tabel pesan digunakan untuk merekam data pesan. Tabel pesan memiliki 6 *field* dengan *primary key*-nya adalah *id_pesan*. Tabel pesan dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	Tindakan
1	<i>id_pesan</i>	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah • Hapus • Lainnya
2	<i>id_peternak</i>	int(11)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
3	<i>id_pakar</i>	int(11)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
4	<i>pengirim</i>	varchar(20)		Tidak	Tidak ada		Ubah • Hapus • Lainnya
5	<i>isi_pesan</i>	text		Tidak	Tidak ada		Ubah • Hapus • Lainnya
6	<i>tanggal</i>	datetime		Tidak	Tidak ada		Ubah • Hapus • Lainnya

Gambar 4.8 Implementasi Tabel Pesan

4.1.9 Implementasi Tabel tmp_penyakit

Tabel tmp_penyakit digunakan untuk merekam data tmp_penyakit. Tabel tmp_penyakit berelasi dengan tabel penyakit dan tabel tmp_penyakit memiliki 2 *field*. Tabel tmp_penyakit dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra Tindakan
1	username	varchar(25)		Tidak	Tidak ada	Ubah Hapus Lainnya
2	kode_penyakit	varchar(6)		Tidak	Tidak ada	Ubah Hapus Lainnya

Gambar 4.9 Implementasi Tabel tmp_penyakit

4.1.10 Implementasi Tabel tmp_gejala

Tabel tmp_gejala digunakan untuk merekam data tmp_gejala. Tabel tmp_gejala berelasi dengan tabel gejala dan tabel tmp_gejala memiliki 3 *field*. Tabel tmp_analisa dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra Tindakan
1	username	varchar(25)		Tidak	Tidak ada	Ubah Hapus Kunci Utama Lainnya
2	kode_gejala	varchar(6)		Tidak	Tidak ada	Ubah Hapus Kunci Utama Lainnya
3	status	enum('1', '0')		Tidak	Tidak ada	Ubah Hapus Kunci Utama Lainnya

Gambar 4.10 Implementasi Tabel tmp_gejala

4.1.11 Implementasi Tabel tmp_analisa

Tabel tmp_analisa digunakan untuk merekam data tmp_analisa. Tabel tmp_analisa berelasi dengan tabel penyakit dan tabel gejala serta tabel tmp_analisa memiliki 3 *field*. Tabel tmp_analisa dapat dilihat pada tabel di bawah ini:

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra Tindakan
1	username	varchar(25)		Tidak	Tidak ada	Ubah Hapus Lainnya
2	kode_penyakit	varchar(6)		Tidak	Tidak ada	Ubah Hapus Lainnya
3	kode_gejala	varchar(6)		Tidak	Tidak ada	Ubah Hapus Lainnya

Gambar 4.11 Implementasi Tabel tmp_analisa

4.2 Implementasi Program

Implementasi sistem pakar mendiagnosis penyakit pada babi dengan metode *forward chaining* ini menggunakan bahasa pemrograman PHP dan dihubungkan dengan *relation database management system MySQL*.

4.2.1 Implementasi Halaman Utama Menu *Home*

Halaman utama aplikasi ini memiliki lima menu yaitu menu *home*, panduan, profil pakar, daftar akun, dan menu *login*. Dalam menu *home* memuat tentang latar belakang masalah yang menjadi alasan pembuatan aplikasi. Tampilan halaman utama menu *home* dapat dilihat pada gambar di bawah ini :

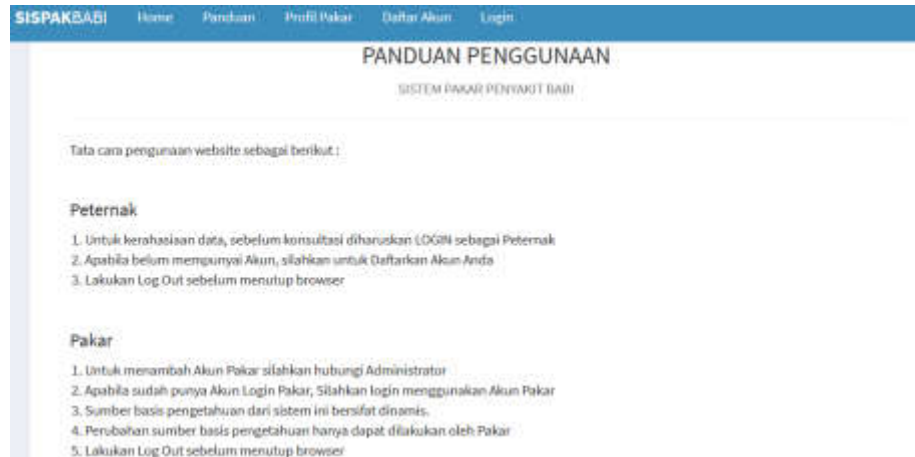


Gambar 4.12 Implementasi Halaman Utama Menu *Home*

4.2.2 Implementasi Halaman Utama Menu Panduan

Halaman utama menu panduan berisi petunjuk penggunaan aplikasi bagi peternak dan pakar, sehingga peternak dan pakar dapat melakukan tugasnya masing-masing.

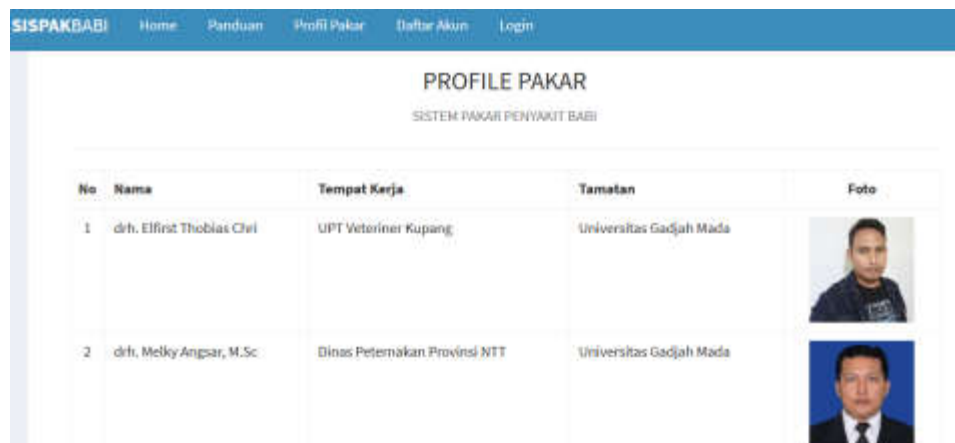
Tampilan halaman utama menu panduan dapat dilihat pada gambar di bawah ini :



Gambar 4.13 Implementasi Halaman Utama Menu Panduan

4.2.3 Implementasi Halaman Utama Menu Profil Pakar

Halaman utama menu profil pakar berisi tentang profil dari dokter hewan. Pengguna dapat mengetahui nama, tempat kerja, tamatan, dan foto dari dokter hewan yang menjadi pakar dalam aplikasi ini. Tampilan halaman utama menu profil pakar dapat dilihat pada gambar di bawah ini :



Gambar 4.14 Implementasi Halaman Utama Menu Profil Pakar

4.2.4 Implementasi Halaman Utama Menu Daftar Akun

Peternak harus mempunyai akun untuk dapat menggunakan aplikasi. Menu ini menampilkan *field-field* yang harus diisi untuk membuat akun. *Field* yang harus diisi antara lain nama, tempat dan tanggal lahir, jenis kelamin, alamat, *username* dan *password*. Tampilan halaman daftar akun dapat dilihat pada gambar di bawah ini :

Gambar 4.15 Implementasi Halaman Utama Menu Daftar Akun

Tampilan halaman daftar akun pada Gambar 4.15 diimplementasikan dengan kode di bawah ini:

```
<?php
require_once '../database.php';
$db = new Database ();
$data = array ();
$data['username'] = addslashes(trim($_POST['username']));
$data['password'] = md5(addslashes(trim($_POST['password'])));
$data['nama'] = addslashes(trim($_POST['nama']));
$data['jk'] = addslashes(trim($_POST['jk']));
$data['alamat'] = addslashes(trim($_POST['alamat']));
$db -> setTable ('peternak_tb');
$db -> insert ($data);
?>
<script type="text/javascript">
    alert('Pendaftaran berhasil! Silahkan login.');
```

Berdasarkan *script* di atas, peternak melakukan proses *peng-input-an* data untuk registrasi, lalu sistem akan menerima data-data tersebut dan disimpan ke tabel peternak dalam *database*. Sehingga peternak yang sudah melakukan proses registrasi bisa *login* ke dalam sistem.

4.2.5 Implementasi Halaman Utama Menu *Login*

Menu ini menampilkan *field-field* yang harus diisi, seperti *username* dan *password*, serta memilih jenis pengguna. Terdapat tiga pengguna dalam sistem ini, yaitu *admin*, *pakar* dan *peternak*. Tampilan halaman *login* dapat dilihat pada gambar di bawah ini:

Gambar 4.16 Implementasi Halaman Utama Menu *Login*

Tampilan halaman utama menu *login* pada Gambar 4.16 diimplementasikan dengan kode di bawah ini:

```
<?php
session_start();
require_once './database.php';
$db = new Database ();
$user_type = addslashes(trim($_POST['user']));
$username = addslashes(trim($_POST['username']));
$password = addslashes(trim($_POST['password']));
$table = $user_type == "admin" ? "admin_tb" : ($user_type == "pakar" ? "pakar_tb" : ($user_type
== "peternak" ? "peternak_tb" : null));
if(!is_null ($table)) {
    $db->setTable ($table);
    $where = array ();
    $where["username"] = $username;
```

```

$where["password"] = md5($password);
$fetch = $db -> fetchRows ($where);
if($db -> getRows () > 0) {
    $_SESSION['login']['type'] = $user_type;
    $_SESSION['login']['data'] = $fetch[0];
    header ('location: ../index.php');
} else {
    $_SESSION['fail']['username'] = $username;
    $_SESSION['fail']['type'] = $user_type;
    $_SESSION['fail']['message'] = "Login gagal. Username atau password tidak
valid.";
    header ('location: ../index.php?page=login');
}
} else
    die ('Ups!! something is wrong. ');
?>

```

Berdasarkan *script login* di atas, saat pengguna melakukan proses *login*, sistem akan mencari data sesuai dengan *input-an username, password* dan tipe *user*. Apabila data yang di-*input*-kan sesuai dengan data pada sistem, maka pengguna akan di alihkan ke halamannya masing-masing. Namun apabila data yang di-*input*-kan tidak sesuai, maka akan muncul pesan *error* dan pengguna akan dialihkan kembali ke halaman *login*.

4.2.6 Implementasi Halaman *Admin Menu Home*

Halaman *admin menu home* berisi tugas-tugas dari *admin* yang dapat dilakukan dalam aplikasi. *Admin* dapat menghapus data peternak dan laporan hasil diagnosa serta menambah, meng-*edit* dan menghapus data pakar.

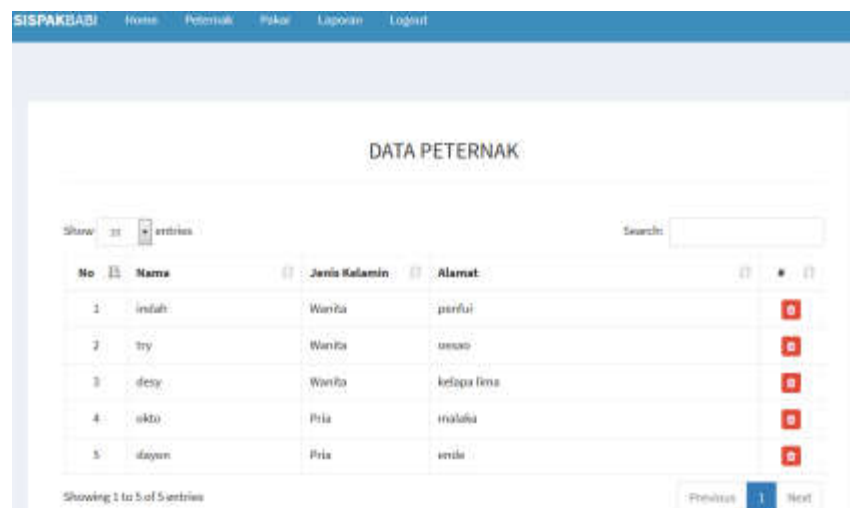
Tampilan halaman *admin* menu *home* dapat dilihat pada gambar di bawah ini:



Gambar 4.17 Implementasi Halaman *Admin* Menu *Home*

4.2.7 Implementasi Halaman *Admin* Peternak

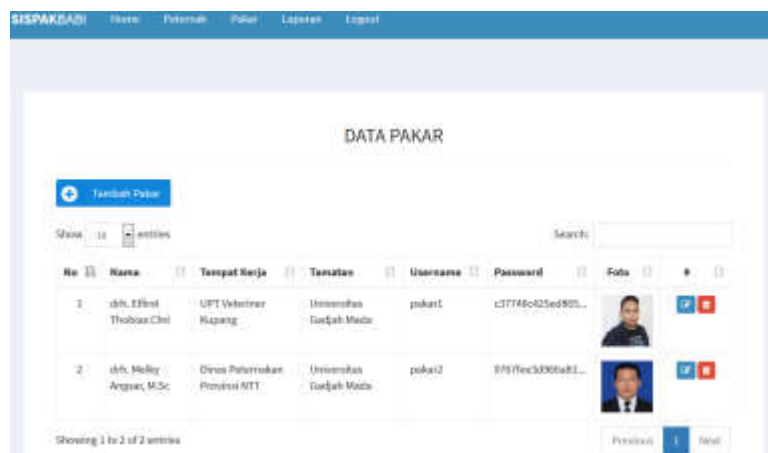
Halaman *admin* menu peternak berisi data peternak yang telah mendaftarkan akun. Data yang dapat dilihat antara lain, nama, jenis kelamin serta alamat. *Admin* berwenang untuk menghapus data peternak. Tampilan halaman *admin* menu peternak dapat dilihat pada gambar di bawah ini :



Gambar 4.18 Implementasi Halaman *Admin* Menu Peternak

4.2.8 Implementasi Halaman *Admin* Menu Pakar

Halaman *admin* menu pakar berisi data pakar. Data pakar yang dapat dilihat antara lain data nama, tempat kerja, tamatan, *username*, *password* dan foto. *Admin* berwenang untuk menambah, meng-*edit* dan menghapus data pakar. Tampilan halaman *admin* menu pakar dapat dilihat pada gambar di bawah ini:



Gambar 4.19 Implementasi Halaman *Admin* Menu Pakar

4.2.9 Implementasi Halaman *Admin* Tambah Data Pakar

Halaman *input* data pakar ini terdapat *field-field* yang harus diisi, untuk menambah data pakar. Data yang harus diisi berupa nama, tempat kerja, tamatan, *username*, *password* dan foto.

Tampilan halaman *admin* tambah data pakar dapat dilihat pada gambar di bawah ini :

Gambar 4.20 Implementasi Halaman *Admin* Tambah Data Pakar

Tampilan halaman tambah data pakar pada Gambar 4.20 diimplementasikan dengan kode di bawah ini:

```
case 'add':
    $nama = addslashes(trim($_REQUEST['nama']));
    $tempat_kerja = addslashes(trim($_REQUEST['tempat_kerja']));
    $tamatan = addslashes(trim($_REQUEST['tamatan']));
    $username = addslashes(trim($_REQUEST['username']));
    $password = md5(addslashes(trim($_REQUEST['password'])));
    $foto = $_FILES['foto']['tmp_name'];
    $nama_foto = $_FILES['foto']['name'];
    move_uploaded_file($foto, "../file/$nama_foto");
    $gambar = $nama_foto;
    $data = array (
        "nama" => $nama,
        "username" => $username,
        "password" => $password,
        "tempat_kerja" => $tempat_kerja,
        "tamatan" => $tamatan,
        "gambar" => $gambar
    );
    $db -> setTable ("pakar_tb");
    $db -> insert ($data);
    echo '
        <script>
            alert("Data berhasil ditambahkan.");
            window.location = "../index.php?page=data-pakar";
        </script> ';
```

Berdasarkan *script* di atas, *admin* melakukan proses tambah data pakar. Data-data yang sudah di-*input* akan disimpan ke dalam tabel pakar dalam *database*, sehingga pakar dapat melakukan proses *login* kedalam sistem.

4.2.10 Implementasi Halaman *Admin* Menu Laporan

Menu ini berisi data peternak yang telah melakukan diagnosa penyakit. Data yang ditampilkan berupa data tanggal diagnosa, nama peternak yang melakukan diagnosa dan hasil diagnosa. *Admin* dapat menghapus data diagnosa. Tampilan halaman *admin* menu laporan dapat dilihat pada gambar di bawah ini :

No	Tanggal	Nama	Hasil Diagnosa
1	2020-01-21	clayen	Hog Cholera
2	2020-01-21	clayen	Hog Cholera

Gambar 4.21 Implementasi Halaman *Admin* Menu Laporan

Tampilan halaman laporan pada Gambar 4.21 diimplementasikan dengan kode di bawah ini:

```
<?php
$id_peternak = $_SESSION['login']['data']['id_peternak'];
$no = 0;
$sql = "SELECT * FROM konsultasi_tb AS kon LEFT JOIN peternak_tb AS pet ON
kon.id_peternak=pet.id_peternak LEFT JOIN penyakit_tb AS pen ON kon.kode_penyakit =
pen.kode_penyakit WHERE kon.id_peternak='$id_peternak' ORDER BY kon.tgl_diagnosa DESC";
foreach ($db->runSql($sql) as $index => $row) {
    $no++;
    extract($row)
    ?>
    <tr>
    <td class="text-center"><?=$no ?></td>
    <td><?=$tgl_diagnosa ?></td>
    <td><?=$nama ?></td>
```

```

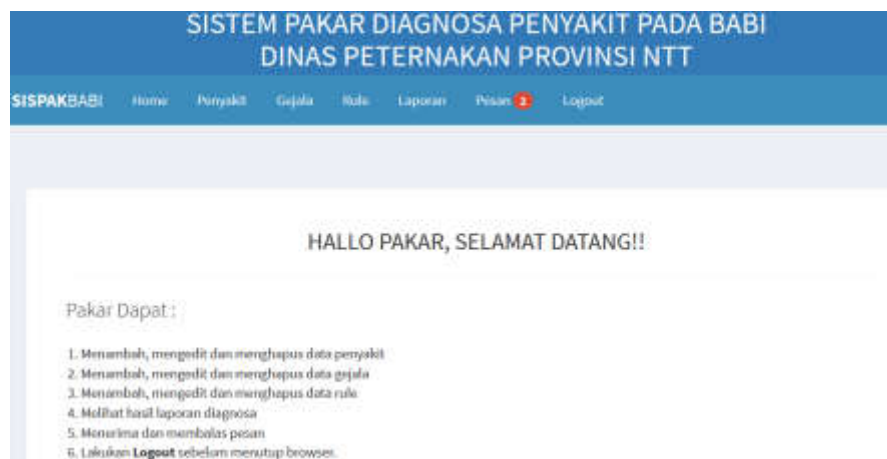
<td><?= $nm_penyakit ?></td>
<td class="text-center">
<a href="action/prs_konsul.php?action=delete&id=<?= $id_konsul ?>" class="btn btn-xs btn-danger
btn-del" alt="Hapus">
<i class="fa fa-trash"></i></a></td></tr>
<?php
}
?>

```

Berdasarkan *script* di atas, *admin* dapat melihat laporan hasil diagnosa, berupa nama penyakit yang diderita, definisi, penyebab serta solusi penanganan penyakit.

4.2.11 Implementasi Halaman Pakar Menu *Home*

Halaman pakar menu *home* berisi tugas-tugas dari pakar yang dapat dilakukan dalam aplikasi. Pakar dapat menambah, meng-*edit* dan menghapus data penyakit, gejala serta *rule*. Pakar dapat melihat hasil diagnosa yang telah dilakukan peternak, serta membaca dan membalas pesan dari peternak. Tampilan halaman pakar menu *home* dapat dilihat pada gambar di bawah ini:

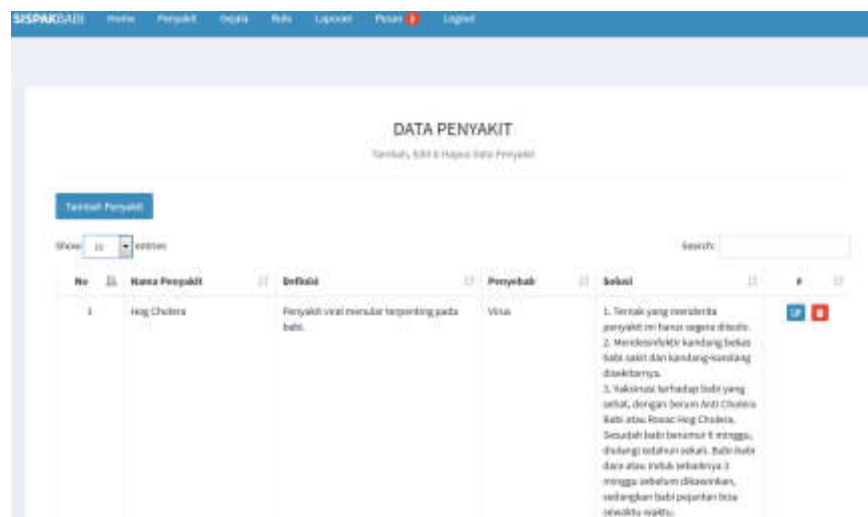


Gambar 4.22 Implementasi Halaman Pakar Menu *Home*

4.2.12 Implementasi Halaman Pakar Menu Penyakit

Halaman pakar menu penyakit berisi tentang data penyakit. Data penyakit berisi kode penyakit, nama penyakit, definisi, penyebab, solusi dan pencegahan penyakit. Pakar berwenang, menambah, meng-*edit* dan menghapus data penyakit.

Tampilan halaman pakar menu penyakit dapat dilihat pada gambar di bawah ini:



Gambar 4.23 Implementasi Halaman Pakar Menu Penyakit

4.2.13 Implementasi Halaman Pakar Tambah Penyakit

Pakar dapat menambah data penyakit pada halaman ini sesuai kebutuhan. Terdapat beberapa *field* yang harus diisi seperti kode penyakit, nama penyakit, definisi, penyebab serta solusi dan pencegahan.

Tampilan halaman pakar tambah penyakit dapat dilihat pada gambar di bawah ini:

Gambar 4.24 Implementasi Halaman Pakar Tambah Penyakit

Tampilan halaman tambah penyakit pada Gambar 4.24 diimplementasikan dengan kode di bawah ini:

```
case 'add':
    extract($_POST);
    $data = array (
        "kode_penyakit" => addslashes(trim($kode_penyakit)),
        "nm_penyakit" => addslashes(trim($nm_penyakit)),
        "penyebab" => addslashes(trim($penyebab)),
        "definisi" => nl2br(addslashes(trim($definisi))),
        "solusi" => nl2br(addslashes(trim($solusi)))
    );
    $db = new Database ();
    $db -> setTable ('penyakit_tb');
    $db -> insert ($data);
    echo '
        <script>
            alert("Data berhasil ditambahkan.");
            window.location = "../index.php?page=data-penyakit";
        </script>
    '
```

Berdasarkan *script* di atas, pakar melakukan tambah data penyakit. Data-data yang sudah di-*input* akan disimpan kedalam tabel penyakit dalam *database*.

4.2.14 Implementasi Halaman Pakar Menu Gejala

Halaman pakar menu gejala berisi tentang data gejala. Data gejala berisi kode gejala dan nama gejala. Pakar dapat, menambah, meng-*edit* dan menghapus data gejala. Tampilan halaman pakar menu gejala dapat dilihat pada gambar di bawah ini:

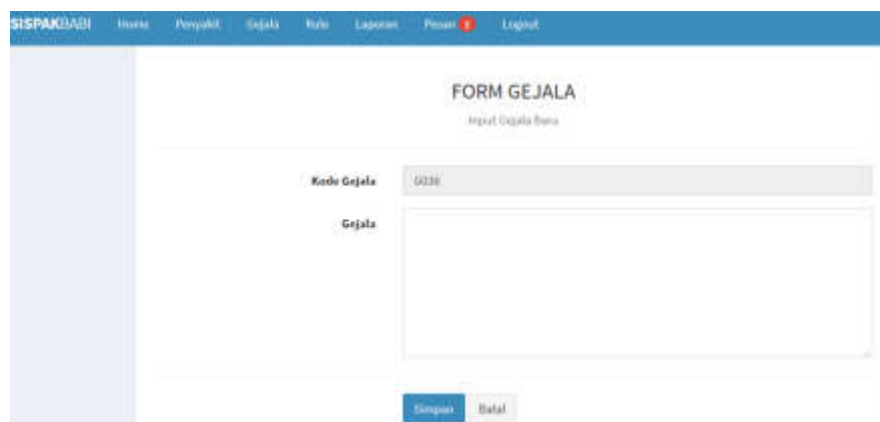


No.	Kode Gejala	Gejala	
1.	0001	Bersakit	 
2.	0002	Meradang pada mata	 
3.	0003	Perkembangan lambat	 
4.	0004	Terdapat yang sering dan lemah karena laki-laki tidak makan sayur-minumnya banyak	 
5.	0005	Buku terlihat bedongman yang menyebabkan buku saling berimpitan	 
6.	0006	Tubuh bagian bawah (sekitar perut) berwarna merah keunguan	 
7.	0007	Siang, ketika beresnya atau atau atau beresnya hingga keputih-putihan	 
8.	0008	Muntah	 

Gambar 4.25 Implementasi Halaman Pakar Menu Gejala

4.2.15 Implementasi Halaman Pakar Tambah Gejala

Pakar dapat menambah data gejala pada halaman ini sesuai kebutuhan. Terdapat beberapa *field* yang harus diisi seperti kode gejala dan nama gejala. Tampilan halaman pakar tambah gejala dapat dilihat pada gambar di bawah ini:



FORM GEJALA
Input Gejala Baru

Kode Gejala: 0030

Gejala:

Simpan Batal

Gambar 4.26 Implementasi Halaman Pakar Tambah Gejala

Tampilan halaman tambah gejala pada Gambar 4.26 diimplementasikan dengan kode di bawah ini:

```
case 'add':
    extract($_POST);
    $data = array (
        "kode_gejala" => addslashes(trim($kode_gejala)),
        "nm_gejala" => addslashes(trim($nm_gejala))
    );
    $db = new Database ();
    $db -> setTable ('gejala_tb');
    $db -> insert ($data);
    echo '
        <script>
            alert("Data berhasil ditambahkan.");
            window.location = "../index.php?page=data-gejala";
        </script>
    ';
```

Berdasarkan *script* di atas, pakar melakukan tambah data gejala. Data-data yang sudah di-*input* akan disimpan kedalam tabel gejala dalam *database*.

4.2.16 Implementasi Halaman Pakar Menu *Rule*

Halaman pakar menu *rule* berisi tentang data rule. Data *rule* berisi basis pengetahuan yang didapat dari data penyakit dan data gejala yang disimpan. Relasi antara penyakit dengan gejala dapat diatur sesuai dengan pengetahuan pakar. Pakar berwenang, menambah, meng-*edit* dan menghapus data *rule*. Tampilan halaman pakar menu *rule* dapat dilihat pada gambar di bawah ini:

No	Kode Gejala	Ya	Tidak	x
1	G001	G002	G003	CF
2	G002	G004	G014	CF
3	G004	P001	G005	CF
4	G005	G006	G012	CF

Gambar 4.27 Implementasi Halaman Pakar Menu *Rule*

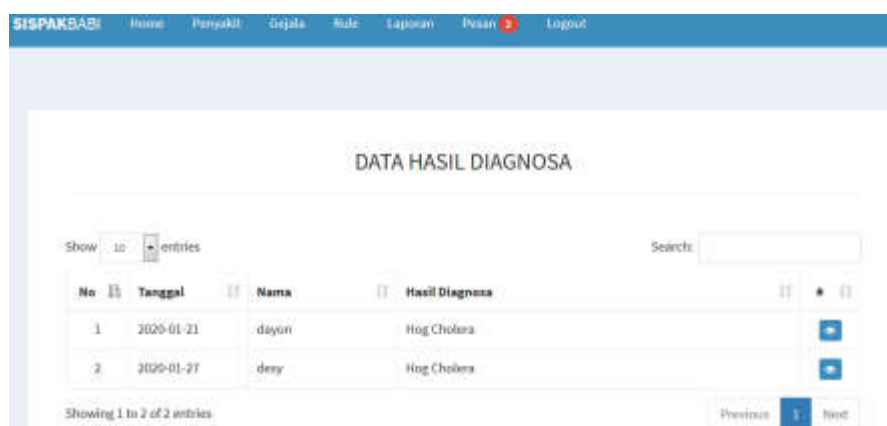
Tampilan halaman tambah *rule* pada Gambar 4.27 diimplementasikan dengan kode di bawah ini:

```
case 'add':
    extract($_POST);
    $data = array (
        "kode_gejala" => addslashes(trim($kode_gejala)),
        "ya" => addslashes(trim($ya)),
        "tidak" => addslashes(trim($tidak))
    );
    $db = new Database ();
    $db -> setTable ('rule_tb');
    $db -> insert ($data);
    echo '
        <script>
            alert("Data berhasil ditambahkan.");
            window.location = "../index.php?page=data-rule";
        </script>
    ';
```

Berdasarkan *script* di atas, pakar dapat melakukan tambah data *rule*. Data-data yang sudah di-*input* akan disimpan kedalam tabel *rule* dalam *database*.

4.2.17 Implementasi Halaman Pakar Menu Laporan

Menu ini berisi data peternak yang telah melakukan diagnosa penyakit. Data yang ditampilkan berupa tanggal diagnosa, nama peternak, serta hasil diagnosa. Tampilan halaman pakar menu laporan dapat dilihat pada gambar di bawah ini:



No	Tanggal	Nama	Hasil Diagnosa
1	2020-01-21	dayon	Hog Cholera
2	2020-01-27	deny	Hog Cholera

Gambar 4.28 Implementasi Halaman Pakar Menu Laporan

Tampilan halaman laporan pada Gambar 4.28 diimplementasikan dengan kode di bawah ini:

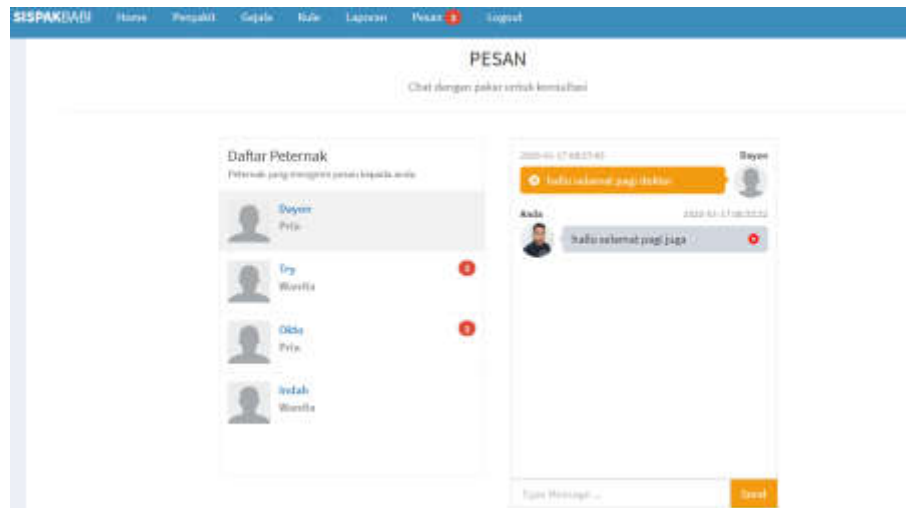
```
<?php
$id_peternak = $_SESSION['login']['data']['id_peternak'];
$no = 0;
$sql = "SELECT * FROM konsultasi_tb AS kon LEFT JOIN peternak_tb AS pet ON
kon.id_peternak=pet.id_peternak LEFT JOIN penyakit_tb AS pen ON kon.kode_penyakit
= pen.kode_penyakit WHERE kon.id_peternak='$id_peternak' ORDER BY
kon.tgl_diagnosa DESC";
foreach ($db -> runSql ($sql) as $index => $row) {
    $no++;
    extract($row)
    ?>
    <tr>
    <td class="text-center"><?=$no ?></td>
    <td><?=$tgl_diagnosa ?></td>
    <td><?=$nama ?></td>
    <td><?=$nm_penyakit ?></td>
    <td class="text-center">
    <a href="action/prs_konsul.php?action=delete&id=<?=$id_konsul ?>" class="btn btn-xs
    btn-danger btn-del" alt="Hapus">
    <i class="fa fa-trash"></i></a></td></tr>
    <?php
    }
    ?>
```

Berdasarkan *script* di atas, pakar dapat melihat laporan hasil diagnosa, berupa nama penyakit yang diderita, definisi, penyebab serta solusi penanganan penyakit.

4.2.18 Implementasi Halaman Pakar Menu Pesan

Menu ini berisi pesan yang dikirim oleh peternak. Pakar dapat membaca dan membalas pesan dari peternak.

Tampilan halaman pakar menu pesan dapat dilihat pada gambar di bawah ini:



Gambar 4.29 Implementasi Halaman Pakar Menu Pesan

4.2.19 Implementasi Halaman Peternak Menu *Home*

Halaman menu *home* ini berisi tugas-tugas yang dapat dilakukan oleh peternak. Peternak dapat melakukan konsultasi, mengirim dan menerima pesan, serta dapat melihat laporan hasil diagnosa. Tampilan halaman peternak menu *home* dapat dilihat pada gambar di bawah ini:



Gambar 4.30 Implementasi Halaman Peternak Menu *Home*

4.2.20 Implementasi Halaman Peternak Menu Diagnosa

Halaman menu diagnosa berfungsi untuk memulai proses konsultasi dengan cara menampilkan pertanyaan dan pilihan gejala pada peternak, dimana pilihan peternak nantinya akan mengarah kepada pertanyaan selanjutnya atau menghasilkan kesimpulan suatu penyakit. Setiap pilihan gejala peternak akan dibandingkan dengan data yang dimiliki oleh sistem. Tampilan halaman peternak menu diagnosa dapat dilihat pada gambar di bawah ini:



Gambar 4.31 Implementasi Halaman Peternak Menu Diagnosa

Tampilan halaman konsultasi pada Gambar 4.31 diimplementasikan dengan kode di bawah ini:

```
<?php
$jwb = isset($_REQUEST['jwb']) ? $_REQUEST['jwb'] : 'G001';
if (preg_match('/G0/', $jwb)) {
    $sql = "SELECT * FROM rule_tb WHERE kode_gejala = '$jwb'";
    $qry = $db->runSql($sql);
    $fch = $qry[0];
    $ops_y = $fch['ya'];
    $ops_t = $fch['tidak'];
    $sql = "SELECT * FROM gejala_tb WHERE kode_gejala = '$jwb'";
    $qry = $db->runSql($sql);
    $fch = $qry[0];
    $stny = $fch['nm_gejala'];
} else if($jwb == "E") {
?>


<i class="fa fa-exclamation-triangle"></i>&nbsp;&nbsp;&nbsp;Maaf sistem tidak dapat menemukan
penyakit dari gejala yang pilih. <br>
Apakah gejala yang pilih sudah tepat?<br><br>
<a href="index.php?page=diagnosa" class="btn btn-success btn-lg">Diagnosa Ulang</a></div>


```

```

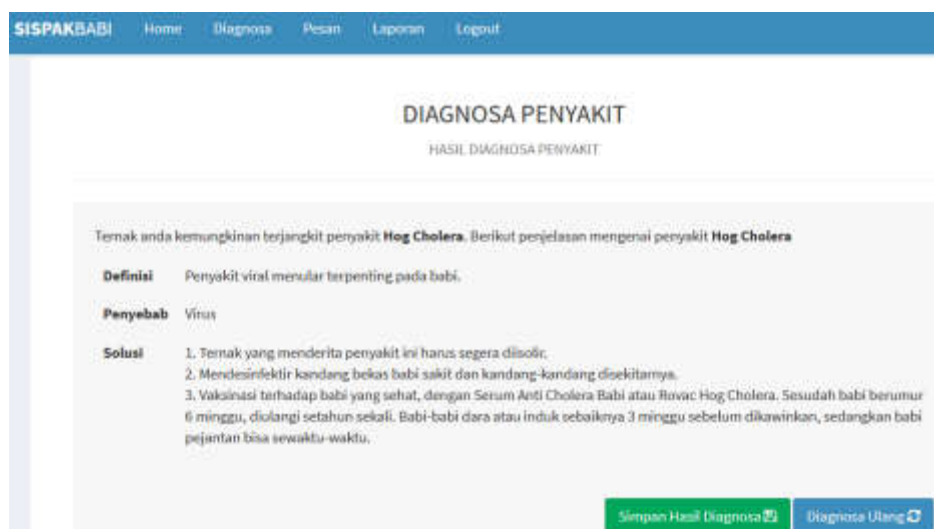
<?php
} else if(preg_match('/P0/', $jwb)) {
    $sql = "SELECT * FROM penyakit_tb WHERE kode_penyakit = '$jwb'";
    $qry = $db->runSql($sql);
    $fch = $qry[0];
    extract($fch);
    $id_peternak = $_SESSION['login']['data']['id_peternak'];
    $tanggal = date('Y-m-d');
    $_SESSION["hasil"][$id_peternak] = $id_peternak;
    $_SESSION["hasil"][$kode_penyakit] = $kode_penyakit;
    $_SESSION["hasil"][$tanggal] = $tanggal;
?>

```

Berdasarkan *script* di atas, peternak melakukan proses konsultasi. Sistem akan memberikan beberapa pertanyaan. Setelah peternak menjawab semua pertanyaan, sistem melakukan analisis berdasarkan data yang ada pada tabel *rule*, sehingga *output*-nya adalah kemungkinan penyakit yang diderita.

4.2.21 Implementasi Halaman Peternak Hasil Diagnosa Penyakit

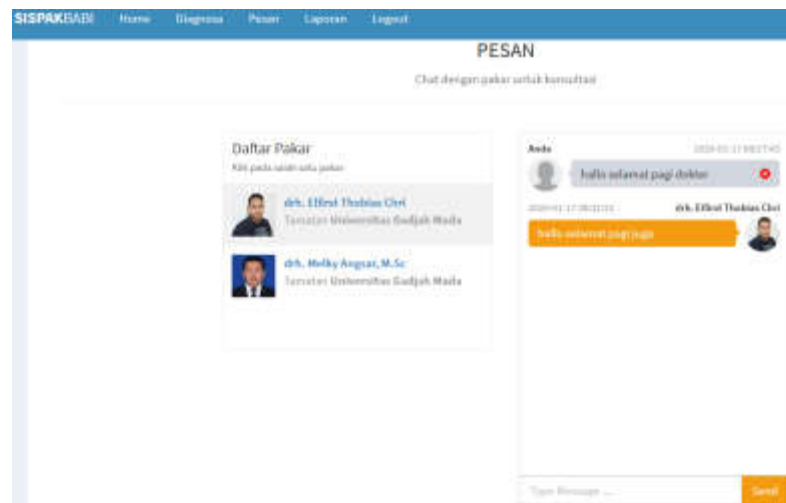
Halaman ini berisi hasil diagnosa penyakit, didalamnya terdapat data nama dan definisi dari penyakit yang diderita, serta penyebab dan solusi penyakit. Peternak dapat menyimpan hasil diagnosa. Tampilan halaman peternak hasil diagnosa penyakit dapat dilihat pada gambar di bawah ini:



Gambar 4.32 Implementasi Halaman Peternak Hasil Diagnosa Penyakit

4.2.22 Implementasi Halaman Peternak Menu Pesan

Halaman ini berisi tampilan untuk mengirim pesan. Peternak dapat memilih pakar, dan langsung mengirimkan pesan. Peternak juga dapat melihat balasan pesan dari pakar. Tampilan halaman peternak menu pesan dapat dilihat pada gambar di bawah ini:



Gambar 4.33 Implementasi Halaman Peternak Menu Pesan

Tampilan halaman pesan pada Gambar 4.33 diimplementasikan dengan kode di bawah ini:

```
case "add":
    extract($_REQUEST);
    $data = array (
        "id_peternak" => $id_peternak,
        "id_pakar" => $id_pakar,
        "pengirim" => $pengirim,
        "isi_pesan" => $isi_pesan,
        "tanggal" => $tanggal
    );
    $db -> setTable ("pesan_tb");
    $db -> insert ($data);

    echo '
        <script>
        window.location= "../index.php?page=pesan&desc='.$desc.'"; </script>';
```

Berdasarkan *script* di atas, peternak dapat mengirimkan pesan ke salah satu pakar. Semua data pesan akan disimpan dalam tabel pesan.

4.2.23 Implementasi Halaman Peternak Menu Laporan

Halaman ini berisi data hasil diagnosa. Data yang dapat dilihat antara lain, tanggal diagnosa, nama dan hasil diagnosa. Tampilan halaman peternak menu laporan dapat dilihat pada gambar di bawah ini:



No	Tanggal	Nama	Hasil Diagnosa	
1	2020-01-28	dayon	Penyakit Cacing Bulat (Ascariids-Roundworm)	 
2	2020-01-21	dayon	Hog Cholera	 

Gambar 4.34 Implementasi Halaman Peternak Menu Laporan

Tampilan halaman laporan pada Gambar 4.34 diimplementasikan dengan kode di bawah ini:

```
<?php
$id_peternak = $_SESSION['login']['data']['id_peternak'];
$no = 0;
$sql = "SELECT * FROM konsultasi_tb AS kon LEFT JOIN peternak_tb AS pet ON
kon.id_peternak=pet.id_peternak LEFT JOIN penyakit_tb AS pen ON kon.kode_penyakit =
pen.kode_penyakit WHERE kon.id_peternak='$id_peternak' ORDER BY kon.tgl_diagnosa DESC";
foreach ($db -> runSql($sql) as $index => $row) {
    $no++;
    extract($row)
    ?>
    <tr> <td class="text-center"><?= $no ?></td>
    <td><?= $tgl_diagnosa ?></td>
    <td><?= $nama ?></td>
    <td><?= $nm_penyakit ?></td>
    <td class="text-center">
    <a href="action/prs_konsul.php?action=delete&id=<?= $id_konsul ?>" class="btn btn-xs btn-danger
    btn-del" alt="Hapus">
    <i class="fa fa-trash"></i></a></td></tr>
<?php } ?>
```

Berdasarkan *script* di atas, peternak dapat melihat laporan hasil diagnosa, berupa nama penyakit yang diderita, definisi, penyebab serta solusi penanganan penyakit.