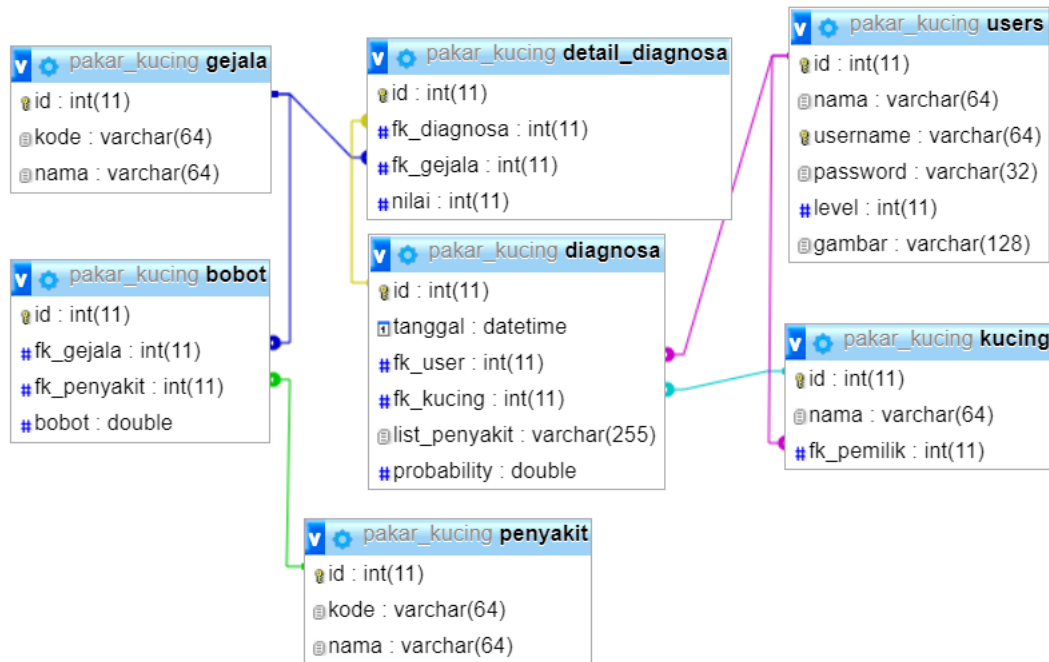


BAB V ANALISIS DAN PERANCANGAN

5.1 Implementasi Database

Berikut ini akan dijelaskan implementasi database yang telah dilakukan pada pembuatan sistem aplikasi ini



Gambar 5.1 Desain Database

Implementasi *database* dengan nama *pakar_kucing* yang memiliki 7 tabel. Tabel tersebut yaitu kucing, diagnosa, gejala, bobot, detail_diagnosa, *users*, penyakit.

5.2 Implementasi Pengkodean Sistem

Aplikasi dibuat menggunakan Bahasa pemrograman *php* dan menggunakan *framework codeigniter*. Implementasi proses pengelompokkan berdasarkan pada analisis dan perancangan terdiri dari beberapa proses yaitu:

1. Mengolah Penyakit

Mengolah penyakit merupakan proses melihat, menambah, mengubah, dan menghapus data penyakit yang terdapat pada *database*. Proses mengolah

penyakit dibagi menjadi beberapa. Berikut kode program ditunjukkan pada tabel Kode Program dibawah ini :

Table 5.1 Kode program load data penyakit

```
public function index()
{
    $data['penyakit_data'] = $this->db
        ->get('penyakit')
        ->result();
    $this->load->view('admin/penyakit/index', $data);
}
```

Berdasarkan pada Kode Program 5.1 load data penyakit, merupakan proses mengambil data penyakit dari *database*.

Table 5.2 Kode program tambah data penyakit

```
$set_penyakit = [
    'kode' => $this->input->post('kode'),
    'nama' => $this->input->post('nama'),
];
$this->db->insert('penyakit', $set_penyakit);
```

Berdasarkan pada Kode Program 5.2 tambah data penyakit, merupakan proses menampilkan data penyakit dari *database*. Proses menambah data penyakit digunakan untuk memasukkan data penyakit kedalam *database*. Proses tersebut dimulai dengan deklarasi *variable* sesuai dengan kolom yang terdapat dalam database. Lalu melakukan perintah *query insert* ke dalam database.

Table 5.3 Kode program mengubah data penyakit

```
$set_penyakit = [
    'kode' => $this->input->post('kode'),
    'nama' => $this->input->post('nama'),
];

$this->db
    ->where('id', $id_penyakit)
    ->update('penyakit', $set_penyakit);
```

Proses mengubah data penyakit untuk mengubah data penyakit kedalam *database*. Proses tersebut dimulai dengan deklarasi *variable* sesuai dengan kolom yang terdapat dalam *database*. Lalu melakukan perintah *query update* ke dalam *database*.

Table 5.4 Kode program hapus data penyakit

```
$this->db
    ->where('id', $id_penyakit)
    ->delete('penyakit');
```

Proses menghapus data penyakit untuk menghapus data penyakit yang pada *database* dengan melakukan perintah *query delete* sesuai dengan *id* yang telah dipilih.

a. Mengolah Gejala

Mengolah gejala merupakan proses melihat, menambah, mengubah, dan menghapus data gejala pada *database*. Beberapa proses mengolah gejala yaitu:

Table 5.5 Kode program menambah gejala

```
$set_gejala = [
    'kode' => $this->input->post('kode'),
    'nama' => $this->input->post('nama'),
];
$this->db->insert('gejala', $set_gejala);
```

Proses menambah gejala digunakan untuk menambahkan gejala kedalam *database*. Proses menambah gejala dengan melakukan perintah *query insert* ke dalam *database*.

Table 5.6 Kode program update gejala

```
$set_gejala = [
    'kode' => $this->input->post('kode'),
    'nama' => $this->input->post('nama'),
];

$this->db
    ->where('id', $id_gejala)
    ->update('gejala', $set_gejala);
```

Proses mengubah gejala digunakan untuk mengubah gejala kedalam *database*. Proses tersebut dengan melakukan perintah *query update* ke dalam *database*.

Table 5.7 Kode program hapus gejala

```
$this->db
->where('id', $id_gejala)
->delete('gejala');
```

Proses menghapus gejala digunakan untuk menghapus data gejala pada *database* dengan cara melakukan perintah *query delete* sesuai dengan id yang telah dipilih.

Table 5.8 Kode program load gejala

```
public function index()
{
    $data['gejala_data'] = $this->db
        ->get('gejala')
        ->result();
    $this->load->view('admin/gejala/index', $data);
}
```

Proses load gejala digunakan untuk melihat data gejala terdiri dari menampilkan tampilan halaman dan mendapatkan data gejala dari *database*.

b. Mengolah Bobot

Berikut proses untuk menambahkan data bobot kedalam *database* :

Table 5.9 Kode program mengolah bobot

```
public function update()
{
    $this->db->truncate('bobot');
    foreach ($this->input->post('bobot') as $id_gejala =>
$value) {
        foreach ($value as $id_penyakit => $nilai) {
            $set = [
                'fk_gejala' => $id_gejala,
                'fk_penyakit' => $id_penyakit,
                'bobot' => $nilai,
            ];
        }
    }
}
```

```

        $this->db->insert("bobot", $set);
    }
}

```

a. Implementasi Diagnosa

Melakukan penjualan merupakan proses menambah data penjualan yang terdapat pada database. Proses melakukan penjualan dibagi menjadi beberapa potongan yaitu:

Table 5.10 Kode program menghitung rata-rata

```

foreach ($bobot as $key => $value) {
    $bobot[$key]['bobot'] =
array_sum($bobot[$key]['bobot']) / count($bobot[$key]['bobot']);
}

```

Pada proses ini dilakukan perhitungan rata-rata jika terjadi kemungkinan 2 penyakit dalam 1 gejala.

Table 5.11 Kode program proses melakukan diagnosa

```

$db_diagnosa = $this->db->get('diagnosa')->result();
    foreach ($db_diagnosa as $key => $value) {
        $db_diagnosa_detail = $this->db->where('fk_diagnosa',
$value->id)->where('nilai', 1)->get("detail_diagnosa")->result();
        $data = [];
        foreach ($db_diagnosa_detail as $k => $v) {
            $data[] = $v->fk_gejala;
        }

        $demstersafer = $this->dempstershafer->process($bobot,
$data);

        $largest = [];
        foreach ($demstersafer as $k => $v) {
            if ($v['is_largest']) {
                $largest = $v;
            }
        }
    }
}

```

Pada tahap ini dilakukan proses diagnosa dengan mengambil data diagnosa lalu melakukan perhitungan *Dempster-Shafer*.

c. Perhitungan Dempster-Shafer

Table 5.12 Kode program Proses Densitas

```

$den_A[] = $this->getDensitas($data[0]);

        foreach ($data as $key => $gejala) {
            if ($key == 0) continue;

$den_B = $this->getDensitas($gejala);

```

Table 5.13 Kode program nilai *bel* dan *plaus*

```

$total_believe_a = 0;
    foreach ($den_A as $k => $v) {
        $total_believe_a += $v['believe'];
    }
    $plau_a = 1 - $total_believe_a;
    $plau_b = 1 - $den_B['believe'];

```

Pada proses ini digunakan untuk melakukan perhitungan awal dari *dempster-shafer* untuk menentukan nilai *belief* dan *plausibility*.

Table 5.14 Kode program perhitungan *dempster-shafer*

```

$dempster = [];
for ($i = 0; $i < (count($den_A)); $i++) {
    $intersect = array_intersect($den_A[$i]['penyakit'],
    $den_B['penyakit']);

    $dempster[] = [
        'penyakit' => $intersect,
        'nilai' => $den_A[$i]['believe'] * $den_B['believe'],
    ];

    $dempster[] = [
        'penyakit' => $den_A[$i]['penyakit'],
        'nilai' => $den_A[$i]['believe'] * $plau_b,
    ];
}

```

```

$dempster[] = [
    'penyakit' => $den_B['penyakit'],
    'nilai' => $plau_a * $den_B['believe'],
];

$sum_konflik = 0;
$group_demster = [];
foreach ($dempster as $k => $v) {
    if (count($v['penyakit']) == 0) {
        $sum_konflik += $v['nilai'];
        continue;
    }
}

$group_demster[implode(",", $v['penyakit'])][0] = $v['nilai'];
}

$result = [];
foreach ($group_demster as $k => $v) {

    $result[] = [
        'penyakit' => explode(",", $k),
        'believe' => ($sum_konflik != 1 ?
array_sum($v) / (1 - $sum_konflik) : 0),
    ];
}

$den_A = $result;

```

d. Menghitung Persentase

Proses ini dilakukan untuk menghitung persentase dari diagnosa penyakit yang diberikan oleh sistem.

Table 5.15 Kode program menghitung persentase

```

$number_only = [];
foreach ($den_A as $key => $value) {
    $number_only[$key]
    number_format(($value['believe'] * 100), 2);
}

$max_key = array_keys($number_only,
max($number_only))[0];

```

```
$ret = [];
```

```
foreach ($den_A as $key => $value) {  
    $ret[] = [  
        'is_largest' => ($max_key == $key ? true :  
false),  
        'penyakit' => $value['penyakit'],  
        'result' => $number_only[$key],  
    ];  
}  
}
```

Pada proses ini dilakukan untuk menghitung persentase dari probabilitas penyakit yang di derita oleh kucing.

a. Input Diagnosa Dokter

Pada proses ini dokter melakukan input diagnosa nya. Selain sistem dapat melakukan perhitungan dan menghitung hasil diagnosa, dokter dapat melakukan input diagnosa melihat dari gejala yang telah dimasukkan oleh user/pemilik kucing.

Table 5.16 Kode Program Input Diagnosa Dokter

```
public function set_penyakit_by_dokter()  
{  
    user_allow([3]);  
    $id_diagnosa = $this->input->post('id_diagnosa');  
    $set = [  
        'fk_penyakit_dokter' => $this->input->  
>post("fk_penyakit"),  
        'fk_dokter' => $this->session->userlogin['id']  
    ];  
    $this->db->where('id',$id_diagnosa)->  
>update('diagnosa',$set);  
  
    redirect("Diagnosa/detail/" . $id_diagnosa);  
}
```

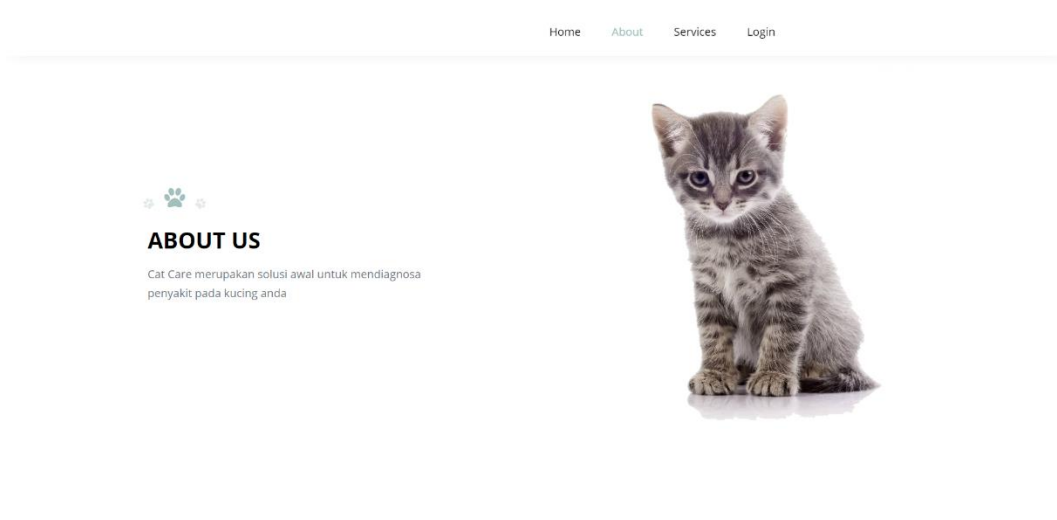
Pada proses ini dokter dapat melihat diagnosa detail dari semua semua pemilik kucing dan juga dapat memasukkan hasil diagnosanya pada sistem.

5.3 Implementasi Tampilan

Implementasi tampilan berdasarkan perancangan antarmuka yang telah dibuat sebelumnya.

1. Tampilan *Home*

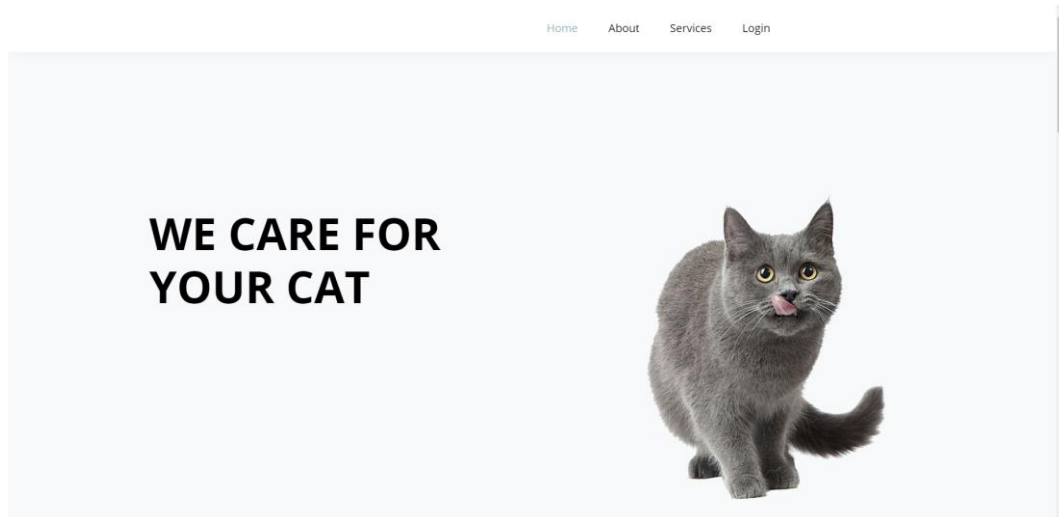
Tampilan users menampilkan halaman awal web sebelum memasuki halaman pemilik kucing/admin. Berikut ini adalah tampilan *Home* yang ditunjukkan pada Gambar 5.2



Gambar 5.2 Tampilan *Home*

2. Tampilan About Us

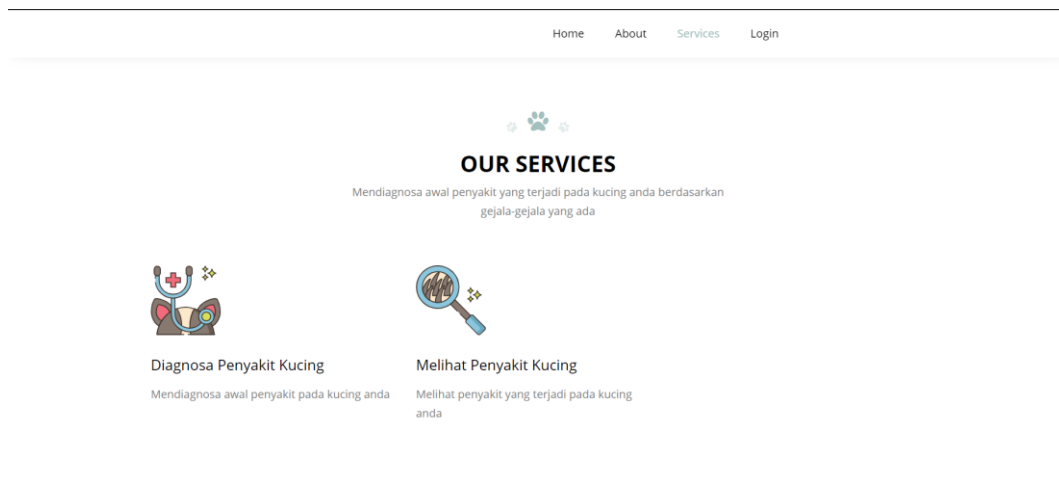
Tampilan About Us menampilkan penjelasan singkat tentang *website* diagnosa kucing.



Gambar 5.3 Tampilan About Us

3. Tampilan Our Service

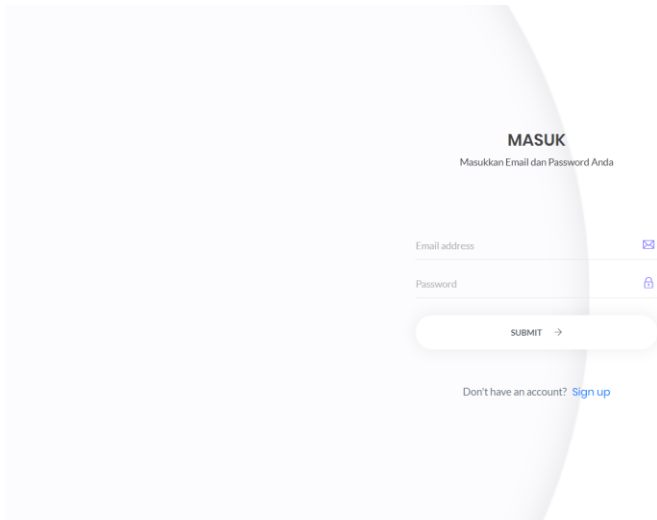
Tampilan Our Service menampilkan tentang fitur yang ada pada website.



Gambar 5.4 Tampilan Our Service

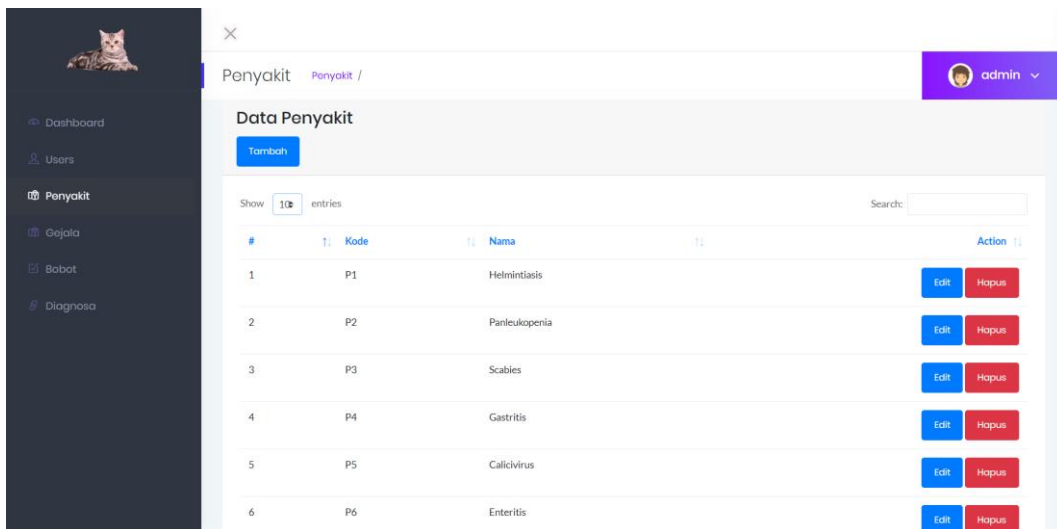
4. Tampilan Login

Tampilan Login merupakan halaman awal sebelum mengakses ke halaman pemilik kucing/admin.



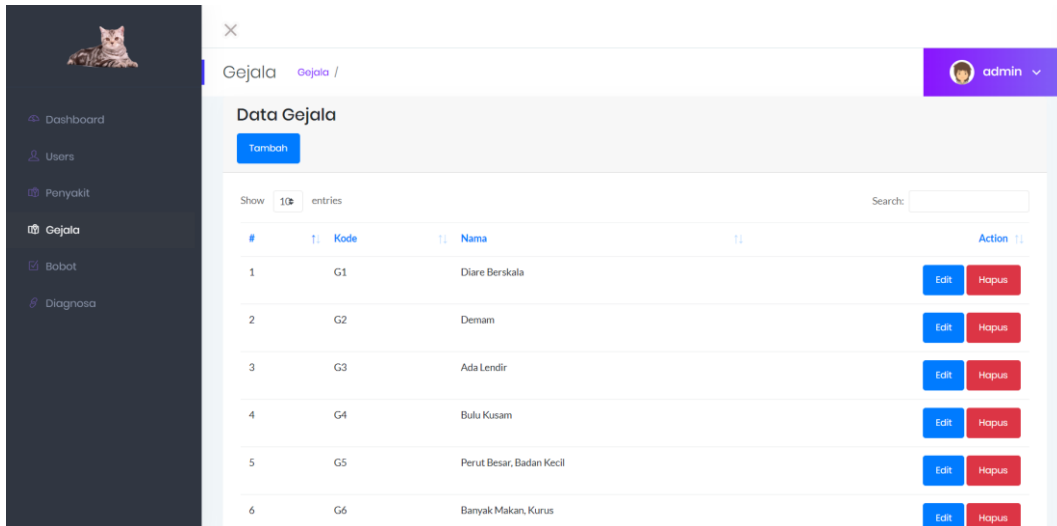
Gambar 5.5 Tampilan login

5. Tampilan Halaman Admin



Gambar 5.6 Tampilan mengolah data penyakit

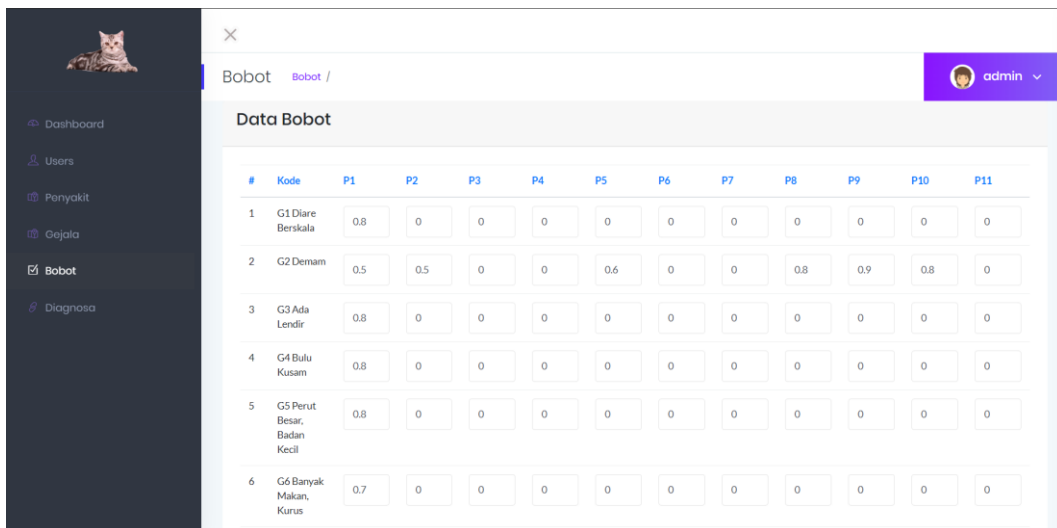
Pada gambar 5.6 menampilkan tampilan mengolah penyakit seperti tambah data penyakit, hapus data dan edit data penyakit yang hanya dapat diakses atau diubah oleh admin.



Gambar 5.7 Tampilan mengolah data gejala

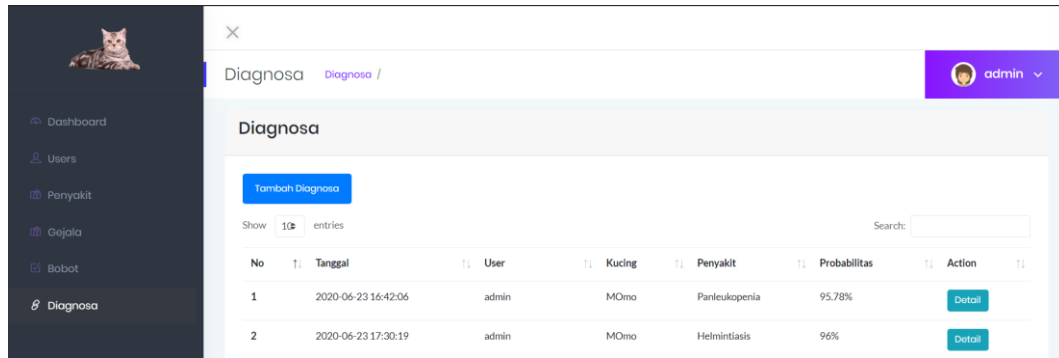
Pada gambar 5.7 menampilkan tampilan gejala menampilkan data gejala. Admin dapat menambahkan data gejala, edit data gejala, dan menghapus data gejala.

Setelah melakukan insert pada gejala akan muncul tampilan daftar gejala pada tampilan bobot yang akan ditunjukkan pada Gambar 5.8 dibawah ini.



Gambar 5.8 Tampilan mengolah bobot

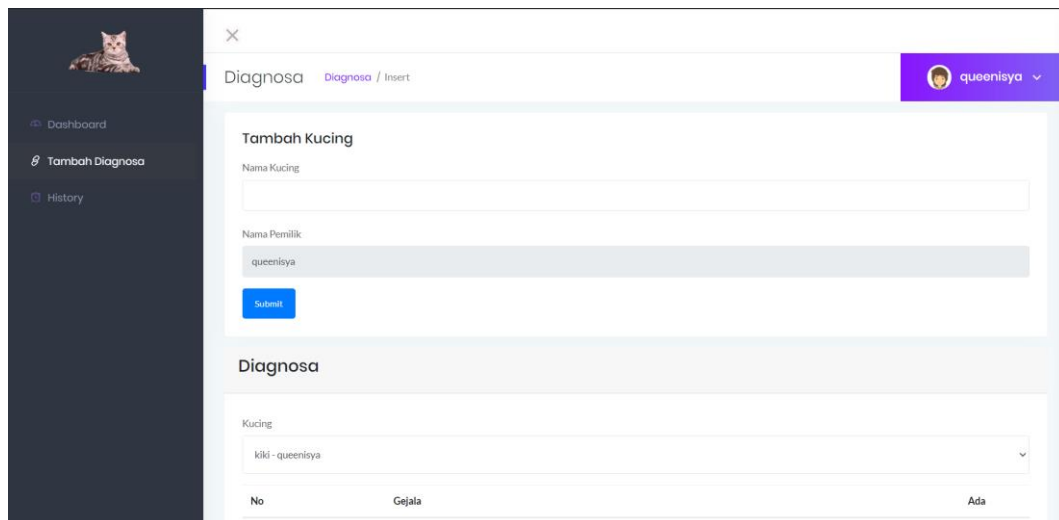
Tampilan diagnosa menampilkan keseluruhan diagnosa penyakit dari pemilik kucing yang ditunjukkan pada gambar 5.9 dibawah ini:



Gambar 5.9 Tampilan data diagnosa

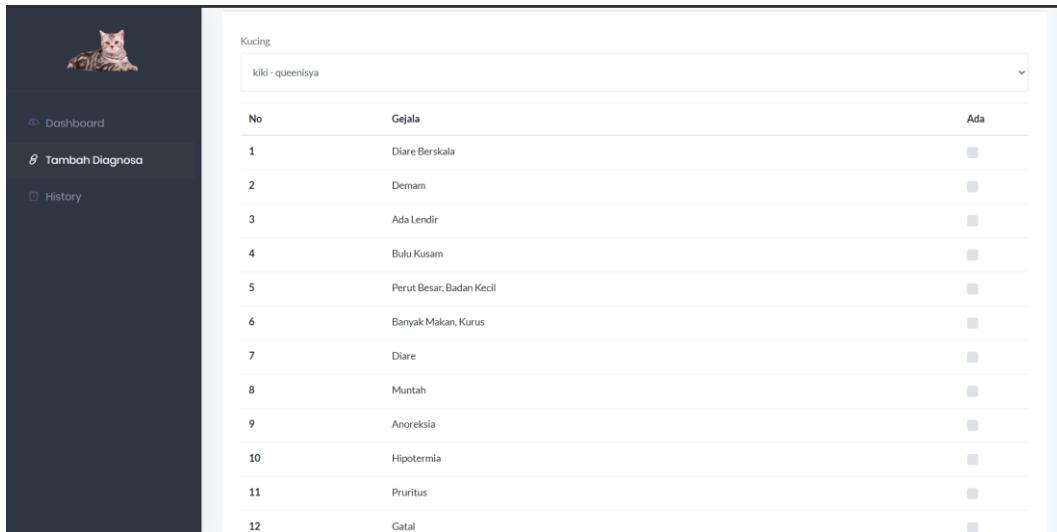
6. Tampilan Halaman Users

Tampilan halaman users menampilkan halaman untuk users/pemilik kucing yang berisi tambah diagnosa dan history.

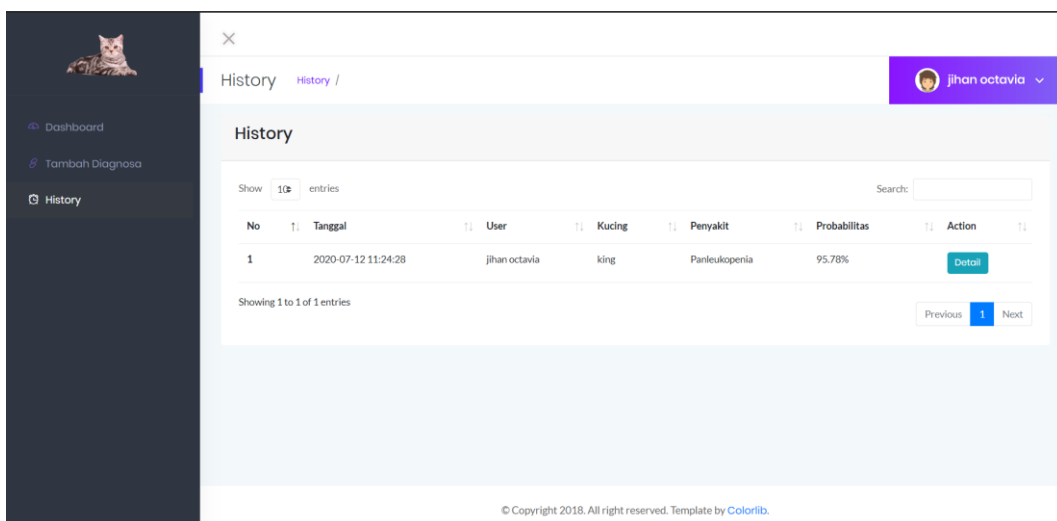


Gambar 5.10 Tampilan tambah diagnosa 1

Pada Gambar 5.10, dapat dilihat pada tampilan tambah diagnose terdapat nama kucing jika user jika pemilik kucing memiliki lebih dari satu kucing. User dapat memberikan tanda *checkbox* pada diagnosa gejala sesuai dengan yang dialami oleh kucing dari pemilik kucing dan akan tampil hasil dari diagnosa.



Gambar 5.11 Tampilan tambah diagnosa 2

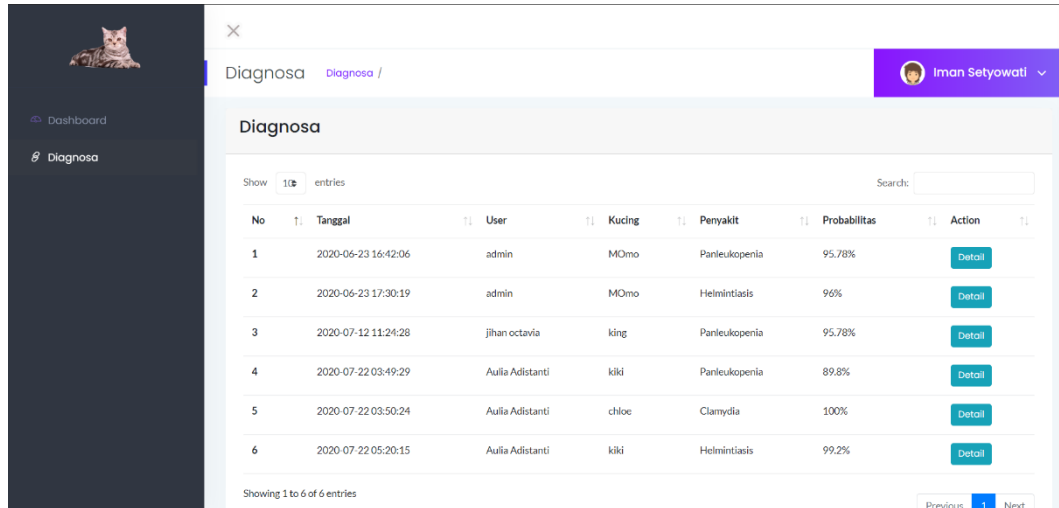


Gambar 5.12 Tampilan history

Tampilan history menampilkan riwayat diagnosa yang pernah dialami oleh kucing dari pemilik kucing.

7. Tampilan halaman dokter

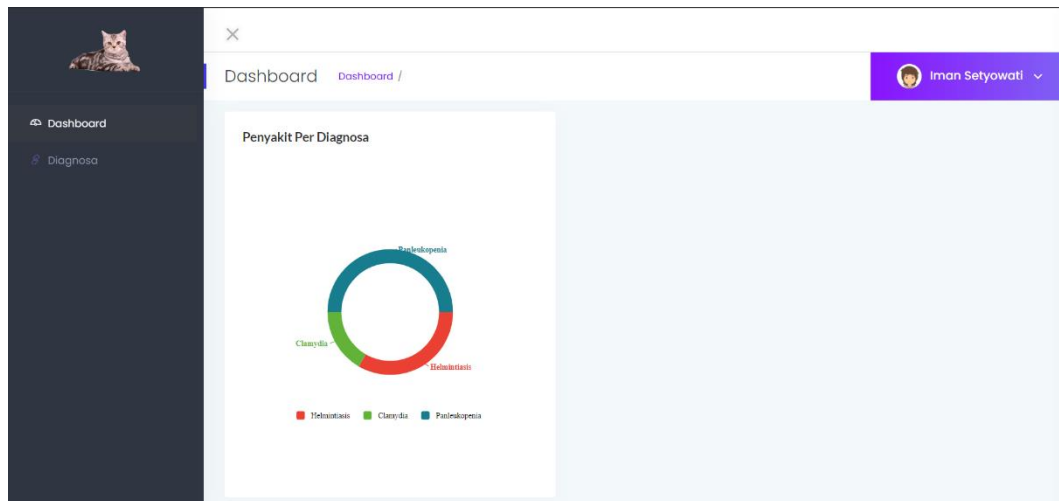
Tampilan halaman dokter menampilkan halaman untuk dokter. Pada halaman ini dokter dapat memberikan diagnosa untuk kucing berdasarkan data pada sistem.



No	Tanggal	User	Kucing	Penyakit	Probabilitas	Action
1	2020-06-23 16:42:06	admin	MOmo	Parneukopenia	95.78%	Detail
2	2020-06-23 17:30:19	admin	MOmo	Helmintiasis	96%	Detail
3	2020-07-12 11:24:28	jihan octavia	kling	Parneukopenia	95.78%	Detail
4	2020-07-22 03:49:29	Aulia Adlistanti	kiki	Parneukopenia	89.8%	Detail
5	2020-07-22 03:50:24	Aulia Adlistanti	chloe	Clamydia	100%	Detail
6	2020-07-22 05:20:15	Aulia Adlistanti	kiki	Helmintiasis	99.2%	Detail

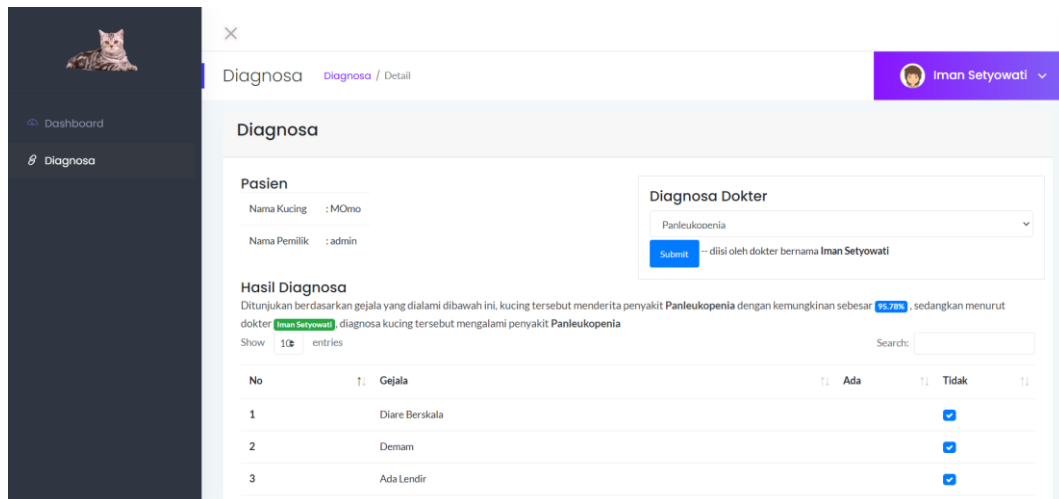
Gambar 5.13 Tampilan Awal Halaman Untuk Dokter

Pada gambar 5.13 menunjukkan halaman terdapat diagram untuk melihat semua diagnosa penyakit per-97z4diagnosa yang ada pada sistem pakar.



Gambar 5.14 Tampilan Awal Halaman untuk Dokter

Pada gambar 5.14 menunjukkan halaman diagnosa untuk melihat semua diagnosa penyakit dari semua user yang ada pada sistem.



Gambar 5.15 Tampilan Detail Diagnosa Dokter

Pada gambar 5.15 menunjukkan halaman diagnosa untuk melihat semua diagnosa penyakit dari semua user yang ada pada sistem dan dokter dapat memasukkan diagnose sesuai dengan analisisnya.

5.4 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem dilakukan dengan cara menjalankan tiap fitur dalam aplikasi dan melihat kesesuaian hasil yang terjadi dengan hasil yang diharapkan.

Table 5.17 Fungsional sistem

No	Usecase	Keterangan	Status
1	Melakukan register	Pemilik kucing dapat melakukan register dan masuk pada halaman home pemilik kucing.	Sesuai
2	Melakukan login	Admin atau pemilik kucing dapat memasuki halaman home setelah melakukan login.	Sesuai
3	Mengolah users	Admin dapat melihat, tambah, edit, hapus pada data user.	Sesuai
4	Mengolah penyakit	Admin dapat melihat, tambah, edit, hapus pada data penyakit.	Sesuai

5	Mengolah gejala	Admin dapat melihat, tambah, edit, hapus pada data gejala.	Sesuai
5	Mengolah bobot	Admin dapat melihat, tambah, edit, hapus pada data bobot.	Sesuai
6	Memasukkan gejala	Pemilik Kucing dapat memasukkan gejala yang dialami oleh kucing peliharaan.	Sesuai
7	Melihat history	Admin/pemilik dapat melihat history gejala.	Sesuai
No	Usecase	Keterangan	Status
1	Mengolah users	Admin dapat melihat, tambah, edit, hapus pada data user.	Sesuai