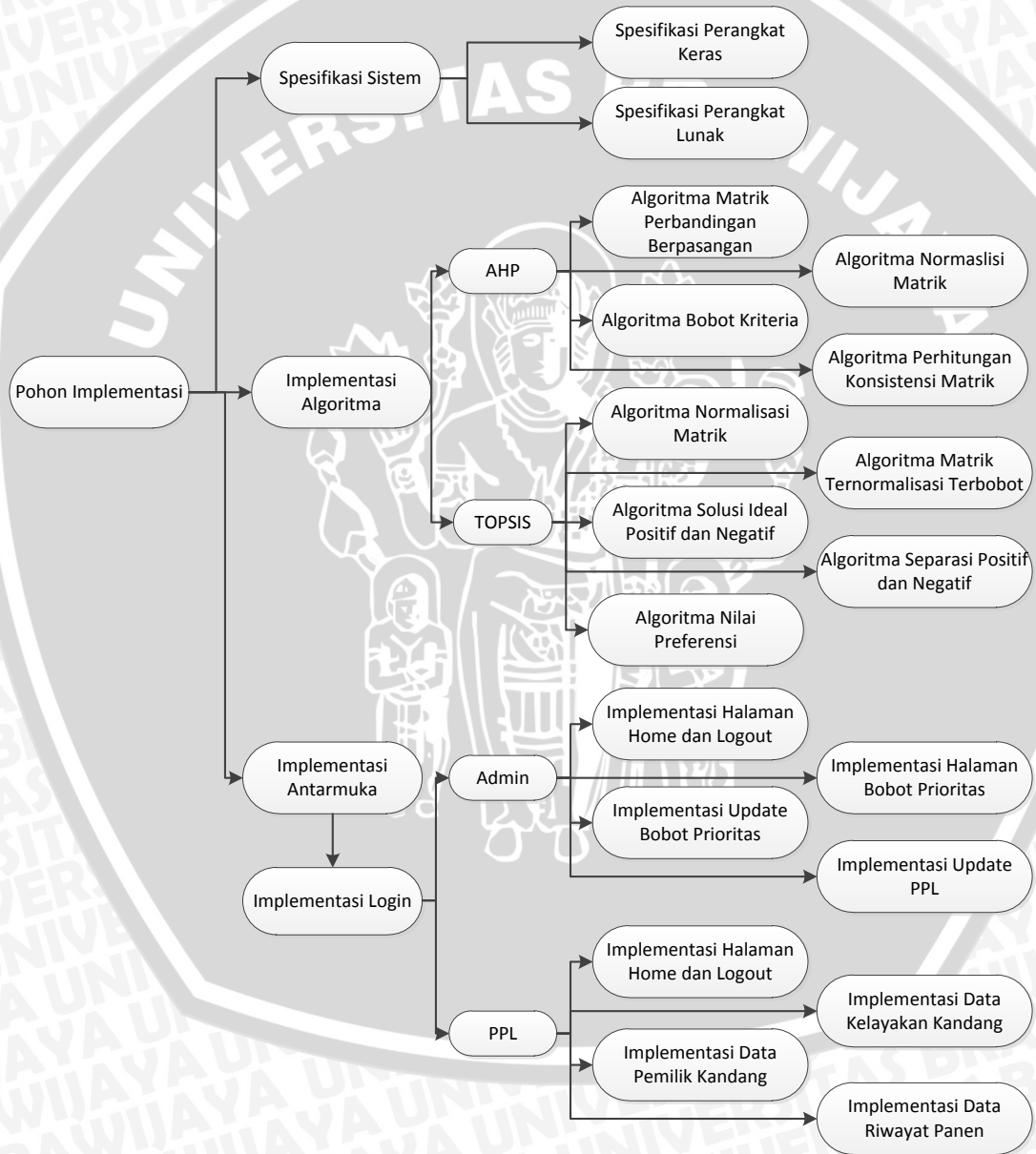


## BAB V IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem pendukung keputusan berdasarkan analisa kebutuhan dan proses perancangan. Pada gambar 5.1 ditunjukkan pohon implementasi sebagai gambaran umum pokok bahasan pada bab 5.



**Gambar 5.1** Pohon Implementasi  
**Sumber :** Implementasi

## 5.1 Spesifikasi Sistem

Spesifikasi sistem yang dibahas meliputi kebutuhan perangkat keras dan perangkat lunak yang digunakan untuk implementasi. Spesifikasi sistem dibahas secara detail agar implementasi berjalan sesuai tujuan.

### 5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras pada sistem pendukung keputusan penentuan kelayakan pengisian bibit ayam broiler dikandang peternak menggunakan komputer yang dijelaskan pada tabel 5.1.

**Tabel 5.1** Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Processor	Intel(R) Core™2 Duo CPU T5870 @ 2.00GHz
Memori (RAM)	2048MB RAM
Chip Type	Mobile Intel(R) 965 Express Chipset Family
DAC Type	Internal
Approx. Total Memory	358MB

**Sumber :** Implementasi

### 5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak pada sistem pendukung keputusan penentuan kelayakan pengisian bibit ayam broiler dikandang peternak menggunakan *software* yang dijelaskan pada tabel 5.2.

**Tabel 5.2** Spesifikasi Perangkat Lunak

<b>Sistem Operasi</b>	Microsoft Windows 7 (32-bit)
<b>Bahasa Pemrograman</b>	HTML 5 dan PHP
<b>Tools Pemrograman</b>	Macromedia Dreamweaver 8
<b>Server Localhost</b>	XAMPP 1.7.3
<b>DBMS</b>	MySQL

**Sumber:** Implementasi

## 5.2 Implementasi Algoritma

Implementasi yang akan dibahas menggunakan bahasa pemrograman php dan menggunakan database *mysql*. Bahasa pemrograman php sendiri digunakan untuk memproses algoritma. Database *mysql* digunakan untuk menyimpan data. Implementasi algoritma ini mengacu pada bab perancangan.

### 5.2.1 AHP

Implementasi algoritma metode AHP ini yang akan digunakan pada user admin yang meliputi pengisian matrik perbandingan. Implementasi algoritma ini meliputi matrik perbandingan berpasangan, normalisasi matrik, Bobot prioritas dan penghitungan konsistensi matrik.

#### 5.2.1.1 Algoritma Matrik Perbandingan Berpasangan

Perhitungan algoritma matrik perbandingan berpasangan ini dilakukan dengan terlebih dahulu mengisi form pada kolom diagonal atas dengan menggunakan skala saaty (1-9). Terdapat 15 data yang nantinya akan dilakukan pembagian setiap nilai yang telah diisi. Gambar 5.2 menjelaskan implementasi algoritma matrik perbandingan berpasangan.

```
1 function kali() {
2   cek();
3   a=eval(form.tinggi_kandang1.value);
4   b=eval(form.kekuatan_kandang1.value);
5   c=eval(form.kelembapan1.value);
6   d=eval(form.jarak_antar_kandang1.value);
7   e=eval(form.keamanan1.value);
8   f=eval(form.kekuatan_kandang2.value);
9   g=eval(form.kelembapan2.value);
10  h=eval(form.jarak_antar_kandang2.value);
11  i=eval(form.keamanan2.value);
12  j=eval(form.kelembapan3.value);
13  k=eval(form.jarak_antar_kandang3.value);
14  l=eval(form.keamanan3.value);
```



```

15 m=eval(form.jarak_antar_kandang4.value);
16 n=eval(form.keamanan4.value);
17 o=eval(form.keamanan5.value);
18 a1=1/a b1=1/b c1=1/c d1=1/d e1=1/e
19 f1=1/f g1=1/g h1=1/h i1=1/i j1=1/j
20 k1=1/k l1=1/l m1=1/m n1=1/n o1=1/o
21 form.riwayat_peternak2.value = a1;
22 form.riwayat_peternak3.value = b1;
23 form.riwayat_peternak4.value = c1;
24 form.riwayat_peternak5.value = d1;
25 form.riwayat_peternak6.value = e1;
26 form.tinggi_kandang3.value = f1;
27 form.tinggi_kandang4.value = g1;
28 form.tinggi_kandang5.value = h1;
29 form.tinggi_kandang6.value = i1;
30 form.kekuatan_kandang4.value = j1;
31 form.kekuatan_kandang5.value = k1;
32 form.kekuatan_kandang6.value = l1;
33 form.kelembapan5.value = m1;
34 form.kelembapan6.value = n1;
35 form.jarak_antar_kandang6.value = o1;
36 form.riwayat_peternak1.value = p;
37 form.tinggi_kandang2.value = q;
38 form.kekuatan_kandang3.value = r;
39 form.kelembapan4.value = s;
40 form.jarak_antar_kandang5.value = t;
41 form.keamanan6.value = u;
42 }

```

**Gambar 5.2** Implementasi Perhitungan Matrik Perbandingan Berpasangan  
**Sumber :** Implementasi

Penjelasan gambar 5.2 implementasi perhitungan matrik perbandingan berpasangan :

- 1) Baris 3-17 adalah mengambil data dari form yang telah diisi pada tabel yang terdapat pada interface pengisian matrik perbandingan berpasangan.
- 2) Baris 18-20 adalah perhitungan pembagian setiap data yang telah dimasukkan tersebut.
- 3) Baris 21-41 adalah penyimpanan data yang telah dihitung tersebut.

### 5.2.1.2 Algoritma Normalisasi Matrik

Algoritma normalisasi matrik dihitung guna untuk mendapatkan penilaian bobot prioritas yang akan dicari. Algoritma ini dihitung dengan melakukan pembagian data setiap kriteria dengan jumlah data setiap kriteria.

```

1  <?php
2  $query=mysql_query("select * from matrik_kriteria");
3  while ($data = mysql_fetch_array($query)){
4  $a=$data['riwayat_peternak1']+$data['riwayat_peternak2']+$data['riwayat_peternak3']
5  +$data['riwayat_peternak4']+$data['riwayat_peternak5']+$data['riwayat_peternak6'];
6  $b=$data['tinggi_kandang1']+$data['tinggi_kandang2']+$data['tinggi_kandang3']
7  +$data['tinggi_kandang4']+$data['tinggi_kandang5']+$data['tinggi_kandang6'];
8  $c=$data['kekuatan_kandang1']+$data['kekuatan_kandang2']+$data['kekuatan_kandang3']+
9  $data['kekuatan_kandang4']+$data['kekuatan_kandang5']+$data['kekuatan_kandang6'];
10 $d=$data['kelembapan1']+$data['kelembapan2']+$data['kelembapan3']+$data['kelembapan4']+
11 $data['kelembapan5']+$data['kelembapan6'];$e=$data['jarak_antar_kandang1']+
12 $data['jarak_antar_kandang2']+$data['jarak_antar_kandang3']+$data['jarak_antar_kandang4']+
13 $data['jarak_antar_kandang5']+$data['jarak_antar_kandang6'];$f=$data['keamanan1']+
14 $data['keamanan2']+$data['keamanan3']+$data['keamanan4']+$data['keamanan5']
15 +$data['keamanan6'];
16 $aa=$data['riwayat_peternak1']/$a; $ab=$data['riwayat_peternak2']/$a;
17 $ac=$data['riwayat_peternak3']/$a; $ad=$data['riwayat_peternak4']/$a;
18 $ae=$data['riwayat_peternak5']/$a; $af=$data['riwayat_peternak6']/$a;
19 $ba=$data['tinggi_kandang1']/$b; $bb=$data['tinggi_kandang2']/$b;
20 $bc=$data['tinggi_kandang3']/$b; $bd=$data['tinggi_kandang4']/$b;
21 $be=$data['tinggi_kandang5']/$b; $bf=$data['tinggi_kandang6']/$b;
22 $ca=$data['kekuatan_kandang1']/$c; $cb=$data['kekuatan_kandang2']/$c;
23 $cc=$data['kekuatan_kandang3']/$c; $cd=$data['kekuatan_kandang4']/$c;
24 $ce=$data['kekuatan_kandang5']/$c; $cf=$data['kekuatan_kandang6']/$c;
25 $da=$data['kelembapan1']/$d; $db=$data['kelembapan2']/$d;
26 $dc=$data['kelembapan3']/$d; $dd=$data['kelembapan4']/$d;
27 $de=$data['kelembapan5']/$d; $df=$data['kelembapan6']/$d;
28 $ea=$data['jarak_antar_kandang1']/$e; $eb=$data['jarak_antar_kandang2']/$e;
29 $ec=$data['jarak_antar_kandang3']/$e; $ed=$data['jarak_antar_kandang4']/$e;
30 $ee=$data['jarak_antar_kandang5']/$e; $ef=$data['jarak_antar_kandang6']/$e;
31 $fa=$data['keamanan1']/$f; $fb=$data['keamanan2']/$f; $fc=$data['keamanan3']/$f;
32 $fd=$data['keamanan4']/$f; $fe=$data['keamanan5']/$f; $ff=$data['keamanan6']/$f; ?>

```

**Gambar 5.3** Implementasi Normalisasi Matrik

**Sumber :** Implementasi

Penjelasan gambar 5.3 implementasi normalisasi matrik :

- 1) Baris 2-3 adalah menampilkan data dari form matrik kriteria.
- 2) Baris 4-15 adalah perhitungan jumlah setiap kriteria matrik perbandingan berpasangan.
- 3) Baris 16-18 adalah perhitungan normalisasi untuk kriteria riwayat peternak.
- 4) Baris 19-21 adalah perhitungan normalisasi untuk kriteria tinggi kandang.
- 5) Baris 22-24 adalah perhitungan normalisasi untuk kriteria kekuatan kandang.
- 6) Baris 25-27 adalah perhitungan normalisasi untuk kriteria kelembapan.
- 7) Baris 28-30 adalah perhitungan normalisasi untuk kriteria jarak antar kandang.
- 8) Baris 31-32 adalah perhitungan normalisasi untuk kriteria keamanan.

### 5.2.1.3 Algoritma Bobot Prioritas

Algoritma perhitungan bobot prioritas didapat dari perhitungan jumlah kolom kekanan dari data yang telah ada. Jumlah kolom tersebut kemudian dibagi dengan jumlah kriteria yang digunakan. Gambar 5.4 menjelaskan algoritma bobot prioritas

```

1  <?php
2  $query=mysql_query("select * from matrik_kriteria");
3  while ($data = mysql_fetch_array($query)){
4  $aaa=$aa+$ba+$ca+$da+$ea+$fa;
5  $bbb=$ab+$bb+$cb+$db+$eb+$fb;
6  $ccc=$ac+$bc+$cc+$dc+$ec+$fc;
7  $ddd=$ad+$bd+$cd+$dd+$ed+$fd;
8  $eee=$ae+$be+$ce+$de+$ee+$fe;
9  $fff=$af+$bf+$cf+$df+$ef+$ff;
10 $aaaa=$aaa/6;
11 $bbbb=$bbb/6;
12 $cccc=$ccc/6;
13 $dddd=$ddd/6;      $eeee=$eee/6;
14 $ffff=$fff/6; }?>

```

**Gambar 5.4** Implementasi Bobot prioritas

**Sumber :** Implementasi



Penjelasan gambar 5.4 implementasi bobot prioritas :

- 1) Baris 4-9 adalah perhitungan jumlah kolom setiap kriteria.
- 2) Baris 10-15 adalah perhitungan bobot kriteria dengan melakukan pembagian dengan jumlah kriteria.

#### 5.2.1.4 Algoritma Perhitungan Konsistensi Matrik

Algoritma perhitungan konsistensi dilakukan ketika pihak admin melakukan perubahan penilaian setiap kriteria. Uji konsistensi dilakukan untuk melihat apakah bobot tersebut layak digunakan atau tidak. Layak bobot kriteria digunakan jika konsistensi tersebut kurang dari sama dengan 0.1. Gambar 5.5 menjelaskan implemmentasi perhitungan konsistensi matrik.

```

1  <?php
   $query=mysql_query("select * from
2  matrik_kriteria");
3  while ($data = mysql_fetch_array($query)){
4  $eigenmax=($a*$aaaa)+($b*$bbbb)+($c*$cccc)
5  +($d*$dddd)+($e*$eeee)+($f*$ffff);
6  $ci=($eigenmax-6)/(6-1);
7  $cr=$ci/1.24;
8  }?>

```

**Gambar 5.5** Implementasi Perhitungan Konsistensi Matrik  
**Sumber :** Implementasi

Penjelasan gambar 5.5 implementasi perhitungan konsistensi matrik :

- 1) Baris 2-3 adalah menampilkan data dari matrik kriteria.
- 2) Baris 4 adalah perhitungan nilai eigen maksimum.
- 3) Baris 6 adalah perhitungan CI atau konsistensi index.
- 4) Baris 7 adalah perhitungan CR atau konsistensi rasio dimana jika nilai konsistensi rasio kurang dari sama dengan 0.1 maka bobot kriteria layak digunakan

## 5.2.2 TOPSIS

Implementasi algoritma metode TOPSIS ini yang akan digunakan pada user PPL yang meliputi normalisasi matrik, perkalian matrik ternormalisasi terbobot, solusi ideal positif dan negatif, separasi positif dan negatif dan nilai preferensi.

### 5.2.2.1 Algoritma Normalisasi Matrik

Normalisasi matrik diambil dari data setiap kandang yang di masukkan kedalam sistem. Data dengan *range* ( 1,3,5 ) yang telah dimasukkan tersebut kemudian dinormalisasi. Gambar 5.6 menunjukkan implementasi normalisasi matrik.

```
1 <?php
2 class topsis{
3     public function normalisasi_matrik($data){
4         //mencari total
5         $total = 0;
6         for($i=0; $i<count($data); $i++)
7         {
8             $total = $total +
9             ($data[$i]*$data[$i]);
10        }
11        // mencari akar
12        $akar =
13        sqrt($total);
14        for($i=0; $i<count($data); $i++)
15        {
16            $result[$i] = $data[$i] /
17            $akar;
18        }
19        return $result;
20    }
21 }
22 ?>
```

**Gambar 5.6** Implementasi Algoritma Normalisasi Matrik  
**Sumber :** Implementasi

Penjelasan gambar 5.6 implementasi algoritma normalisasi matrik :

- 1) Baris 2-18 adalah fungsi untuk menghitung matrik ternormalisasi.



- 2) Baris 5-9 adalah mencari perhitungan total terlebih dahulu setiap kriteria kelayakan kandang.
- 3) Baris 11 adalah mencari akar setelah semua setiap kriteria di total.
- 4) Baris 12-16 adalah perhitungan normalisasi, setiap data kriteria akan dibagi dengan akar dari jumlah yang telah dihitung pada baris 11.

### 5.2.2.2 Algoritma Matrik Ternormalisasi Terbobot

Algoritma matrik ternormalisasi terbobot ini adalah perkalian matrik antara bobot kriteria dan matrik ternormalisasi. Bobot kriteria didapat dari metode AHP dan matrik ternormalisasi didapat dari perhitungan yang terdapat pada gambar 5.6. Gambar 5.7 menunjukkan implementasi matrik ternormalisasi terbobot.

```

1 <?php
2 $query = mysql_query("select * from bobot_prioritas");
3 $data = mysql_fetch_array($query);
4 for($i=0; $i<count($riwayat); $i++){
5     $hasilrp[$i] = $riwayat[$i]*$data['riwayat_peternak'];
6     $hasilrk[$i] = $tinggi[$i]*$data['tinggi_kandang'];
7     $hasilkk[$i] = $kekuatan[$i]*$data['kekuatan_kandang'];
8     $hasilkl[$i] = $kelembapan[$i]*$data['kelembapan'];
9     $hasiljak[$i] =
10     $jarak_antar_kandang[$i]*$data['jarak_antar_kandang'];
11     $hasilkm[$i] = $keamanan[$i]*$data['keamanan'];
12 }
13 ?>

```

**Gambar 5.7** Implementasi Algoritma Matrik Ternormalisasi Terbobot  
**Sumber :** Implementasi

Penjelasan gambar 5.7 implementasi algoritma matrik ternormalisasi terbobot :

- 1) Baris 2-3 adalah menampilkan data bobot prioritas.
- 2) Baris 4-11 adalah perulangan, kemudian setiap data kandang yang telah ternormalisasi dikalikan dengan bobot prioritas.

### 5.2.2.3 Algoritma Solusi Ideal Positif dan Solusi Ideal Negatif

Algoritma solusi ideal positif dan negatif didapat dari perhitungan nilai maksimal dan minimal pada setiap kriteria. Gambar 5.8 menunjukkan implementasi algoritma solusi ideal positif dan solusi ideal negatif.

```
1 <?php
2     $maksimal1 = max($hasilrp);
3     $maksimal2 = max($hasiltk);
4     $maksimal3 = max($hasilkk);
5     $maksimal4 = max($hasilkl);
6     $maksimal5 = max($hasiljak);
7     $maksimal6 = max($hasilkm);
8
9     $minimal1 = min($hasilrp);
10    $minimal2 = min($hasiltk);
11    $minimal3 = min($hasilkk);
12    $minimal4 = min($hasilkl);
13    $minimal5 = min($hasiljak);
14    $minimal6 = min($hasilkm);
15 ?>
```

**Gambar 5.8** Implementasi Algoritma Solusi Ideal Positif dan Solusi Ideal Negatif  
**Sumber :** Implementasi

Penjelasan gambar 5.8 implementasi algoritma solusi ideal positif dan solusi ideal negatif :

- 1) Baris 2-7 adalah pencarian solusi ideal positif dengan menggunakan perhitungan pencarian maksimal setiap kriteria.
- 2) Baris 9-14 adalah pencarian solusi ideal negatif dengan menggunakan perhitungan pencarian minimal setiap kriteria.

### 5.2.2.4 Algoritma Separasi Positif dan Separasi Negatif

Algoritma separasi positif dan separasi negatif adalah pencarian jarak ideal. Separasi positif adalah perhitungan akar dari data jumlah setiap kandang dikurangi dengan maksimal setiap kriteria tersebut. Separasi negatif dihitung dengan akar dari data setiap kandang dikurangi dengan minimal setiap kriteria. Gambar 5.9

menunjukkan implementasi algoritma separasi positif sedangkan gambar 5.10 menunjukkan algoritma separasi negatif.

```

1 <?php
2 for($i=0; $i<count($hasilrp); $i++){
3     $a[$i]=sqrt(($hasilrp[$i]-$maksimal1)*($hasilrp[$i]-$maksimal1))
4     +(($hasiltk[$i]-$maksimal2)*($hasiltk[$i]-$maksimal2))+(($hasilkk[$i]-
5     $maksimal3)*
6     ($hasilkk[$i]-$maksimal3))+(($hasilkl[$i]-$maksimal4)*($hasilkl[$i]-
7     $maksimal4))+
8     (($hasiljak[$i]-$maksimal5)*($hasiljak[$i]-$maksimal5))+(($hasilkm[$i]-
9     $maksimal6)*
10    ($hasilkm[$i]-$maksimal6));
11 }?>

```

**Gambar 5.9** Implementasi Algoritma Separasi Positif  
**Sumber :** Implementasi

Penjelasan gambar 5.9 implementasi algoritma separasi positif :

- 1) Baris 2-8 adalah perulangan tiap jumlah data yang ada
- 2) Baris 3-7 adalah perhitungan separasi positif dengan melakukan perhitungan akar dari data setiap kandang dikurangi dengan nilai maksimal yang didapat dari solusi ideal positif.

```

1 <?php
2 for($i=0; $i<count($hasilrp); $i++){
3     $b[$i]=sqrt(($hasilrp[$i]-$minimal1)*($hasilrp[$i]-
4     $minimal1)+($hasiltk[$i]-$minimal2)*
5     $hasiltk[$i]-$minimal2)+($hasilkk[$i]-$minimal3)*($hasilkk[$i]-
6     $minimal3)+
7     ($hasilkl[$i]-$minimal4)*($hasilkl[$i]-$minimal4)+($hasiljak[$i]-
8     $minimal5)*
9     ($hasiljak[$i]-$minimal5)+($hasilkm[$i]-$minimal6)*($hasilkm[$i]-
10    $minimal6));
11 }?>

```

**Gambar 5.10** Implementasi Algoritma Separasi Negatif  
**Sumber :** Implementasi



Penjelasan gambar 5.10 implementasi algoritma separasi negatif :

- 1) Baris 2-8 adalah perulangan tiap jumlah data yang ada
- 2) Baris 3-7 adalah perhitungan separasi negatif dengan melakukan perhitungan akar dari data setiap kandang dikurangi dengan nilai maksimal yang didapat dari solusi ideal negatif.

### 5.2.2.5 Algoritma Nilai Preferensi

Algoritma nilai preferensi adalah perhitungan terakhir dari metode topsis. Suatu kandang dikatakan layak jika nilai dari preferensi tersebut lebih dari 0.5. Perhitungan nilai preferensi ini adalah dengan pembagian separasi negatif dibagi dengan jumlah separasi negatif dengan separasi positif. Gambar 5.11 menunjukkan algoritma perhitungan nilai preferensi.

```
1 <?php
  $query = mysql_query("select data_kandang.id AS iddatakandang,
2 userkandang.username AS usernames,
  userkandang.alamat_pemilik AS alamat_pemiliks,data_kandang.alamat_kandang AS
3 alamat_kandangs,
  data_kandang.tanggal_masuk AS tanggal_masuks from data_kandang, userkandang
4 where data_kandang.id=userkandang.id order by data_kandang.id asc");
5 while ($data = mysql_fetch_array($query)) {
6   $hasil=$b[$i]/($b[$i]+$a[$i]);
7   if($hasil>"0.5"){
8     echo "layak";
9   }else{
10    }
11    echo "tidak layak";
12  }
13 }?>
```

**Gambar 5.11** Implementasi Algoritma Nilai Preferensi  
**Sumber :** Implementasi

Penjelasan gambar 5.11 implementai algoritma nilai preferensi :

- 1) Baris 2-5 menampilkan data dari kandang.
- 2) Baris 6-12 adalah perhitungan nilai preferensi dimana jika nilai hasil tersebut lenih dari 0.5 maka kandang tersebut dikatakan layak.

### 5.3 Implementasi Antar Muka

Implementasi antarmuka diterapkan berdasarkan pada bab perancangan. Sistem pendukung keputusan ini menggunakan bahasa pemrograman HTML 5 dan PHP. Berikut adalah implementasi antarmuka yang digunakan sistem untuk berinteraksi dengan *end user*.

#### 5.3.1 Implementasi Login

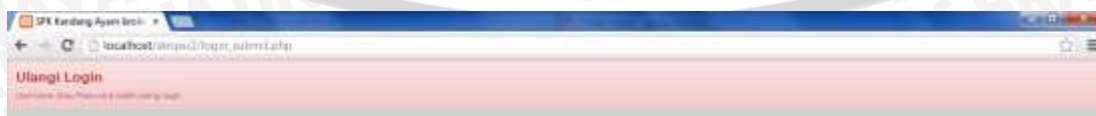
Implementasi halaman login adalah antarmuka yang digunakan user untuk masuk pada sistem. Halaman login harus diisi oleh user agar dapat masuk pada sistem sesuai dengan hak akses user tersebut sebagai admin PPL atau PPL. Gambar 5.12 menunjukkan implementasi halaman login.



**Gambar 5.12** Implementasi Login

**Sumber :** Implementasi

Jika user salah memasukkan username dan juga password maka sistem akan menampilkan pesan warning bahwa username atau password salah. Gambar 5.13 menunjukkan halaman login salah dikarenakan username atau password salah.



**Gambar 5.13** Halaman Login salah

**Sumber :** Implementasi

### 5.3.1.1 Admin

Halaman admin ini hanya dapat diakses oleh user admin. Admin mempunyai hak akses untuk menilai bobot kriteria yang nantinya sebagai acuan bobot kriteria untuk perhitungan kelayakan kandang. Admin sendiri dapat melihat bobot prioritas, edit bobot dan update PPL.

#### 5.3.1.1.1 Implementasi Halaman Home dan Logout

Halaman home admin adalah halaman yang berisi tentang sub menu yang disajikan untuk user admin. Halaman home admin hanya dapat diakses oleh user admin. Halaman home disajikan beberapa sub menu bobot prioritas, edit bobot, update PPL dan juga terdapat logout sistem. Gambar 5.14 menunjukkan implementasi halaman home dan logout sistem.



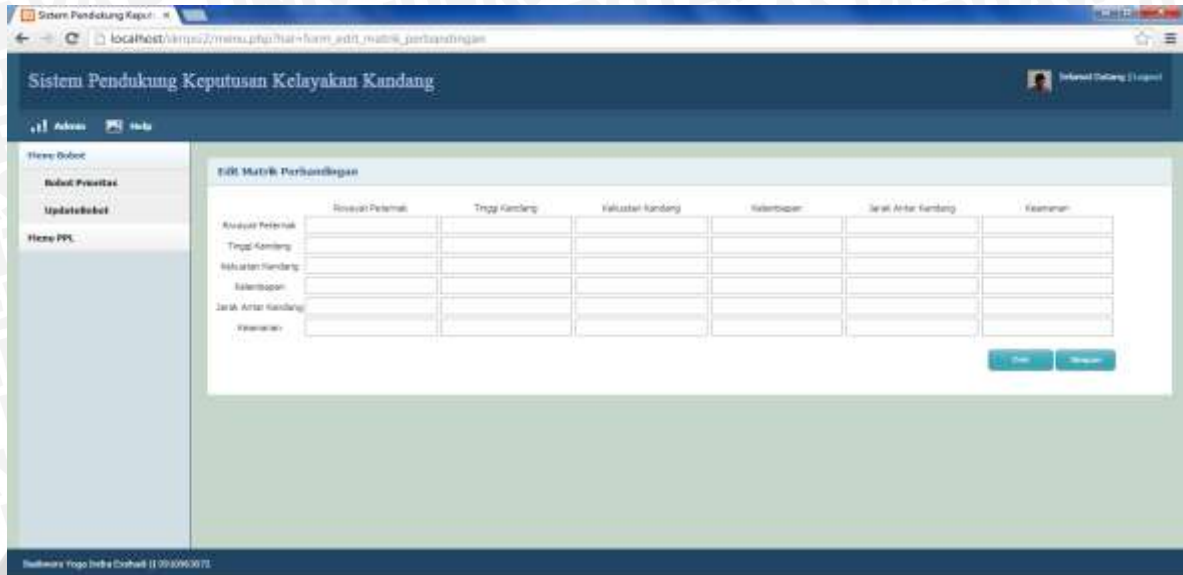
**Gambar 5.14** Implementasi Halaman Home dan Logout  
**Sumber :** Implementasi

#### 5.3.1.1.2 Implementasi Halaman Bobot Prioritas

Implementasi halaman bobot prioritas ini menampilkan nilai bobot prioritas dan juga nilai dari konsistensi yang dihasilkan dari bobot tersebut. Nilai bobot yang



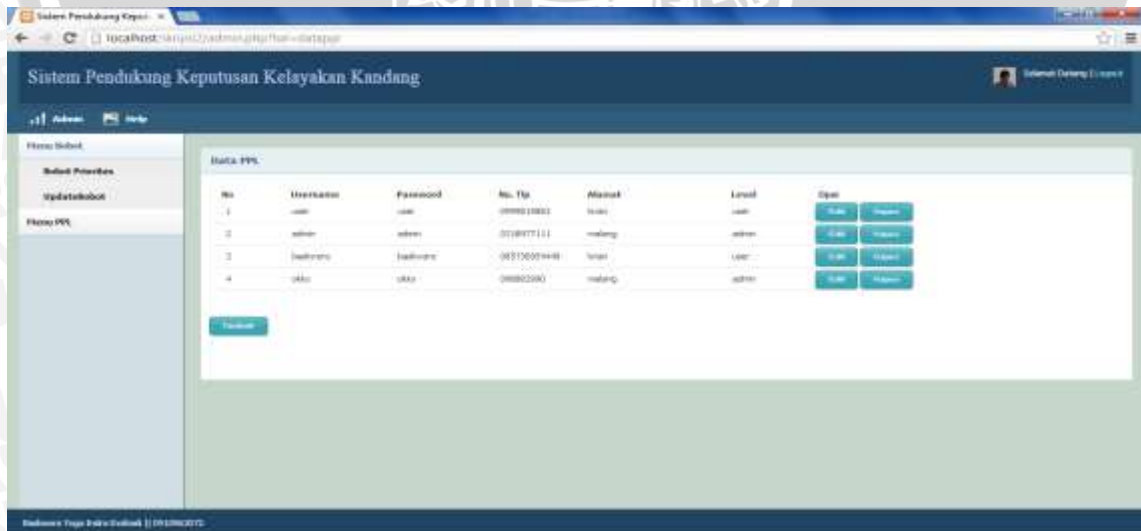




**Gambar 5.16** Implementasi Update Bobot Prioritas  
**Sumber :** Implementasi

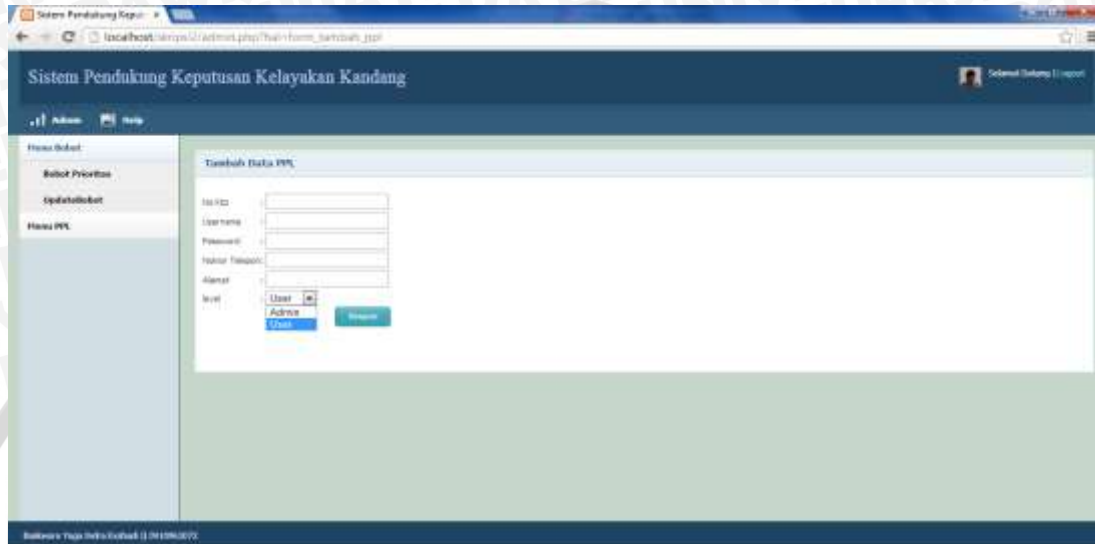
**5.3.1.1.4 Implementasi Update PPL**

Implementasi update PPL yang dibahas ini meliputi tambah PPL, edit PPL serta Hapus PPL. User admin dapat memanajemen user sapa saja yang dapat melakukan penginputan kriteria kandang maupun siapa saja yang bertindak sebagai admin. Gambar 5.17 menunjukkan implementasi tentang update PPL.



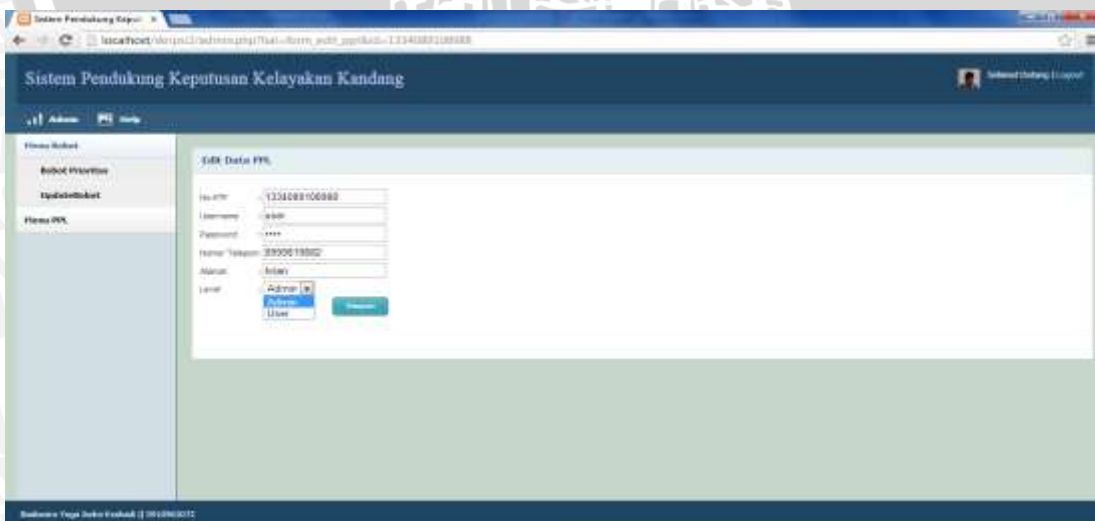
**Gambar 5.17** Implementasi Update PPL  
**Sumber :** Implementasi

User admin dapat melakukan tambah data PPL dengan mengklik tombol tambah pada halaman update PPL. Gambar 5.18 menampilkan halaman tambah PPL.



**Gambar 5.18** Tambah PPL  
**Sumber :** Implementasi

User admin juga dapat melakukan edit data jika ingin mengubah hak akses user ataupun mengubah nama dan alamat user. User admin dapat mengklik tombol edit dan akan disajikan halaman seperti pada gambar 5.19.



**Gambar 5.19** Edit PPL  
**Sumber :** Implementasi



### 5.3.1.2 PPL

User PPL mempunyai hak akses untuk mengupdate data pemilik kandang, data penilaian kandang dan juga data riwayat panen. User PPL ini mempunyai kewajiban untuk melakukan penilaian pada sistem guna untuk mendapatkan nilai kelayakan sebuah kandang.

#### 5.3.1.2.1 Implementasi Halaman *Home* dan *Logout*

Halaman home PPL adalah halaman yang berisi tentang sub menu yang disajikan untuk user PPL. Halaman home PPL hanya dapat diakses oleh user PPL. Halaman home disajikan beberapa sub menu update data pemilik kandang, data kelayakan kandang dan juga data riwayat panen. Gambar 5.20 menunjukkan implementasi halaman home dan logout sistem.

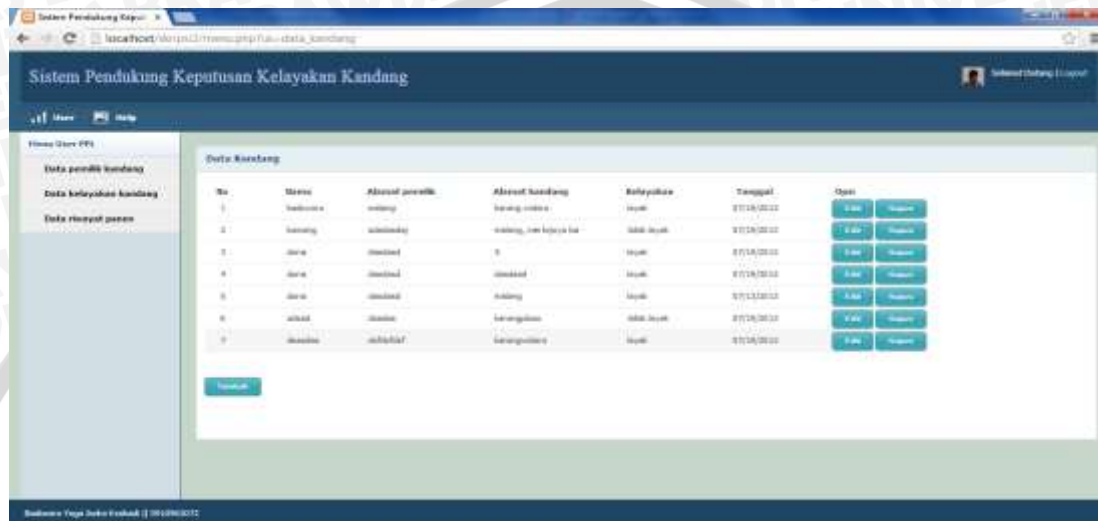


**Gambar 5.20** Impelmentasi Halaman Home dan Logout  
**Sumber :** Implementasi

#### 5.3.1.2.2 Implementasi Data Kelayakan Kandang

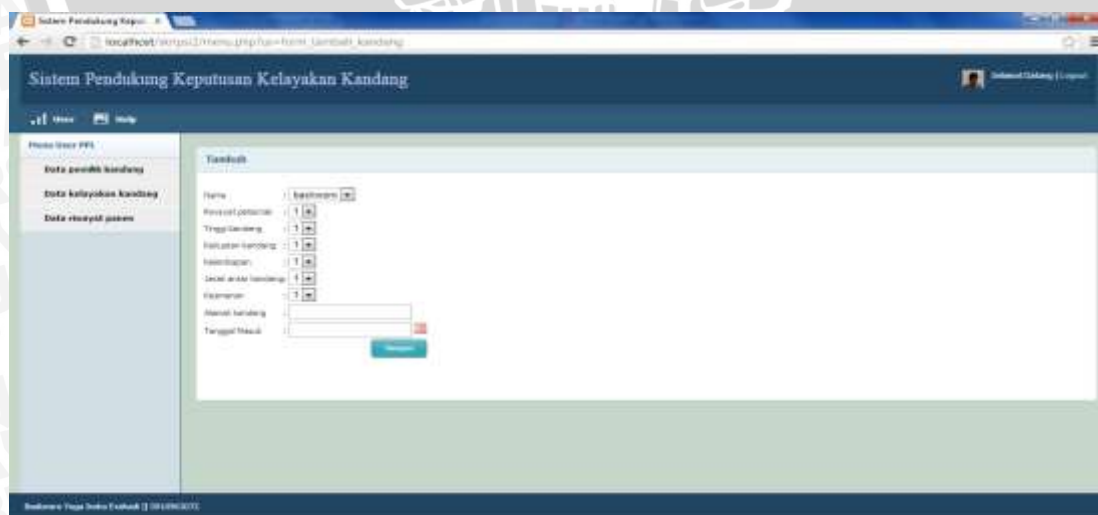
Implementasi data kelayakan kandang merupakan halaman utama dari PPL. Halaman sub menu kelayakan kandang ini yang ditampilkan meliputi nama pemilik

kandang, alamat kandang, alamat pemilik kandang, kelayakan kandang dan juga tanggal kandang tersebut dimasukkan pada sistem. Gambar 5.21 menampilkan tentang implementasi kelayakan kandang.



**Gambar 5.21** Implementasi Kelayakan Kandang  
**Sumber :** Implementasi

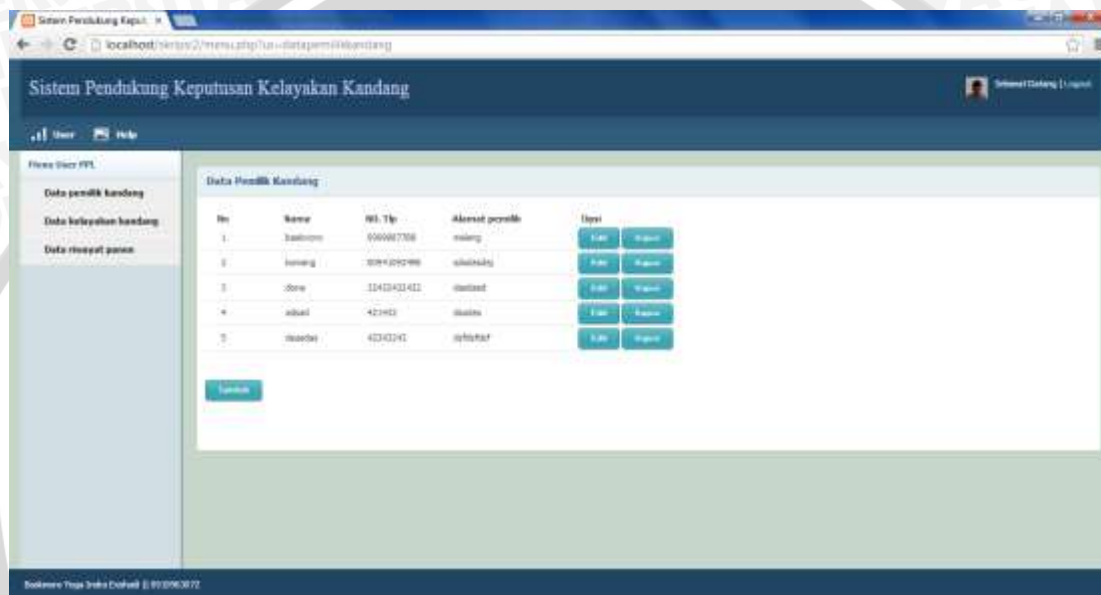
User PPL yang ingin menambahkan atau memasukkan data kandang dapat mengklik tombol tambah. Tampilan tombol tambah kandang dapat dilihat pada gambar 5.22.



**Gambar 5.22** Impelemntasi Tambah Data Kandang  
**Sumber :** Implementasi

### 5.3.1.2.3 Implementasi Data Pemilik Kandang

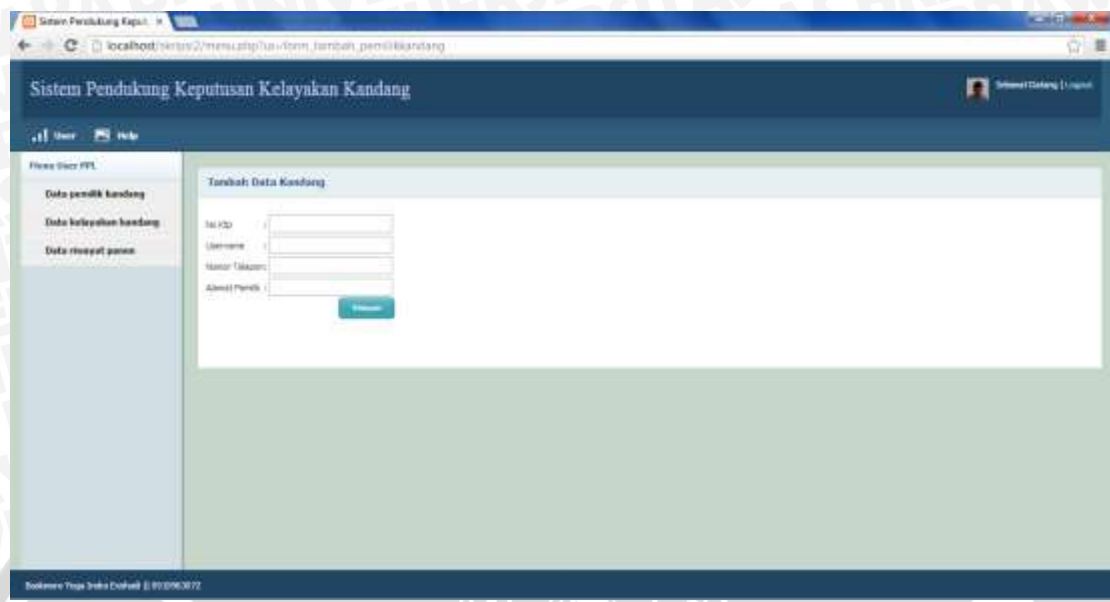
Implementasi data pemilik kandang adalah sub menu yang menyajikan informasi tentang pemilik kandang. Informasi implementasi data pemilik kandang meliputi no ktp pemilik, alamat pemilik dan no telepon pemilik. Gambar 5.23 menampilkan data pemilik kandang.



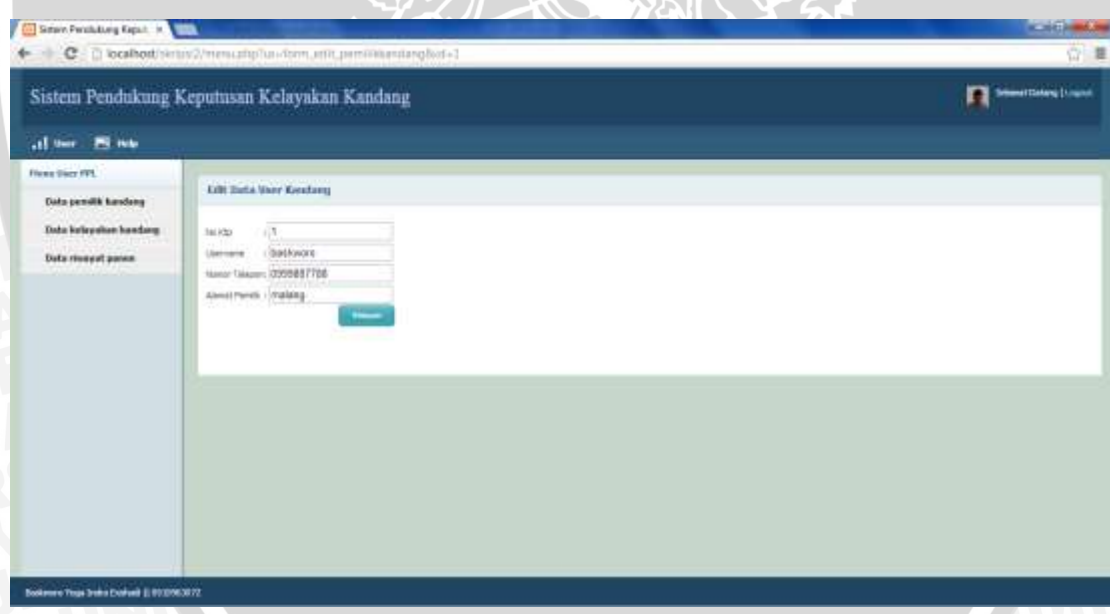
**Gambar 5.23** Implementasi Data Pemilik Kandang  
**Sumber :** Implementasi

User PPL dapat menambah pemilik kandang dengan mengklik tombol tambah pada halaman data pemilik kandang. Gambar 5.24 menampilkan form untuk tambah data pemilik kandang. Pemilik kandang juga dapat mengklik tombol edit jika ingin melakukan edit data pemilik kandang dengan melakukan klik tombol edit pada halaman tersebut. Gambar 5.25 menunjukkan untuk edit data pemilik kandang. User PPL juga mempunyai hak akses untuk melakukan penghapusan pemilik kandang jika pemilik sudah tidak pernah tau keluar dari kemitraan dengan perusahaan.





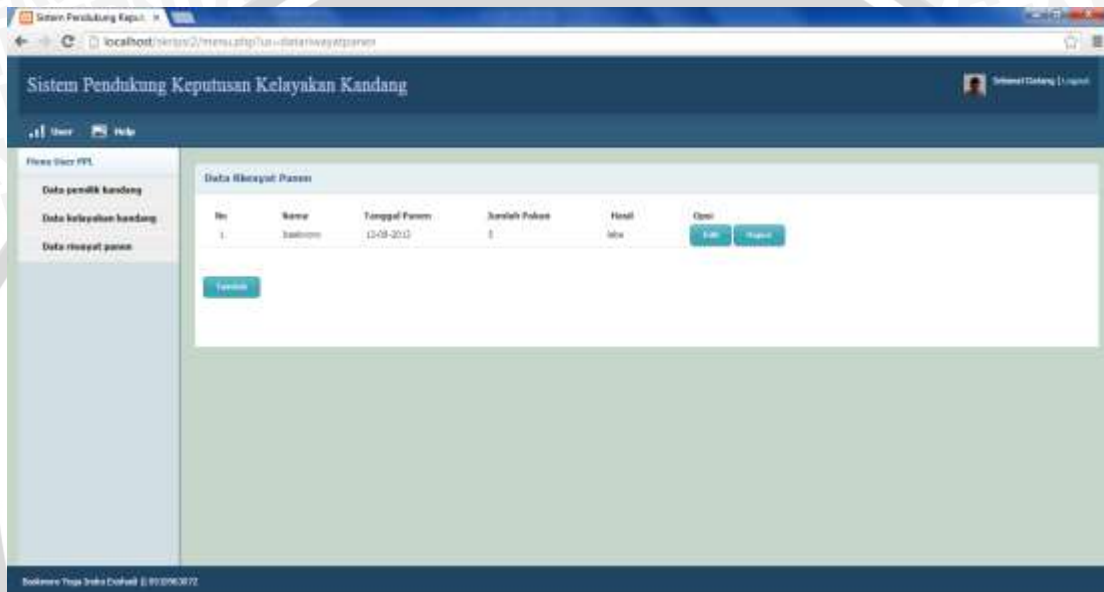
**Gambar 5.24** Implementasi Tambah Data Pemilik Kandang  
**Sumber :** Implementasi



**Gambar 5.25** Implementasi Edit Data Pemilik Kandang  
**Sumber :** Implementasi

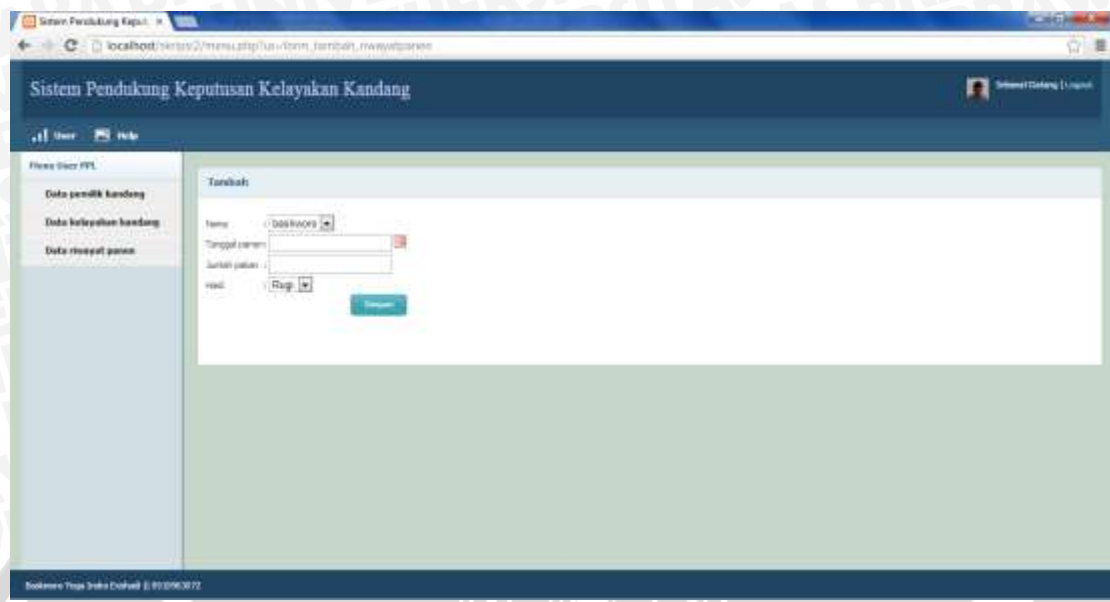
#### 5.3.1.2.4 Implementasi Data Riwayat Panen

Implementasi data riwayat panen dapat digunakan untuk PPL sebagai bahan pertimbangan penilaian riwayat peternak. Data riwayat panen menyimpan data panen peternak apakah mendapatkan laba atau rugi pada setiap panen tiba. Gambar 5.26 menampilkan halaman data riwayat panen.



**Gambar 5.26** Implementasi Data Riwayat Panen  
**Sumber :** Implementasi

User PPL disini juga mempunyai hak akses untuk menambah, mengedit ataupun melakukan penghapusan data riwayat panen seperti pada sub menu lainnya. User PPL disini dapat mengawasi riwayat panen setiap peternak apakah dalam satu periode mendapatkan laba atau rugi. Gambar 5.27 menampilkan halaman untuk menambah data riwayat panen. Gambar 5.28 menampilkan halaman untuk melakukan edit riwayat panen.



**Gambar 5.27** Implemetasi Tambah Data Riwayat Panen  
**Sumber :** Implementasi

