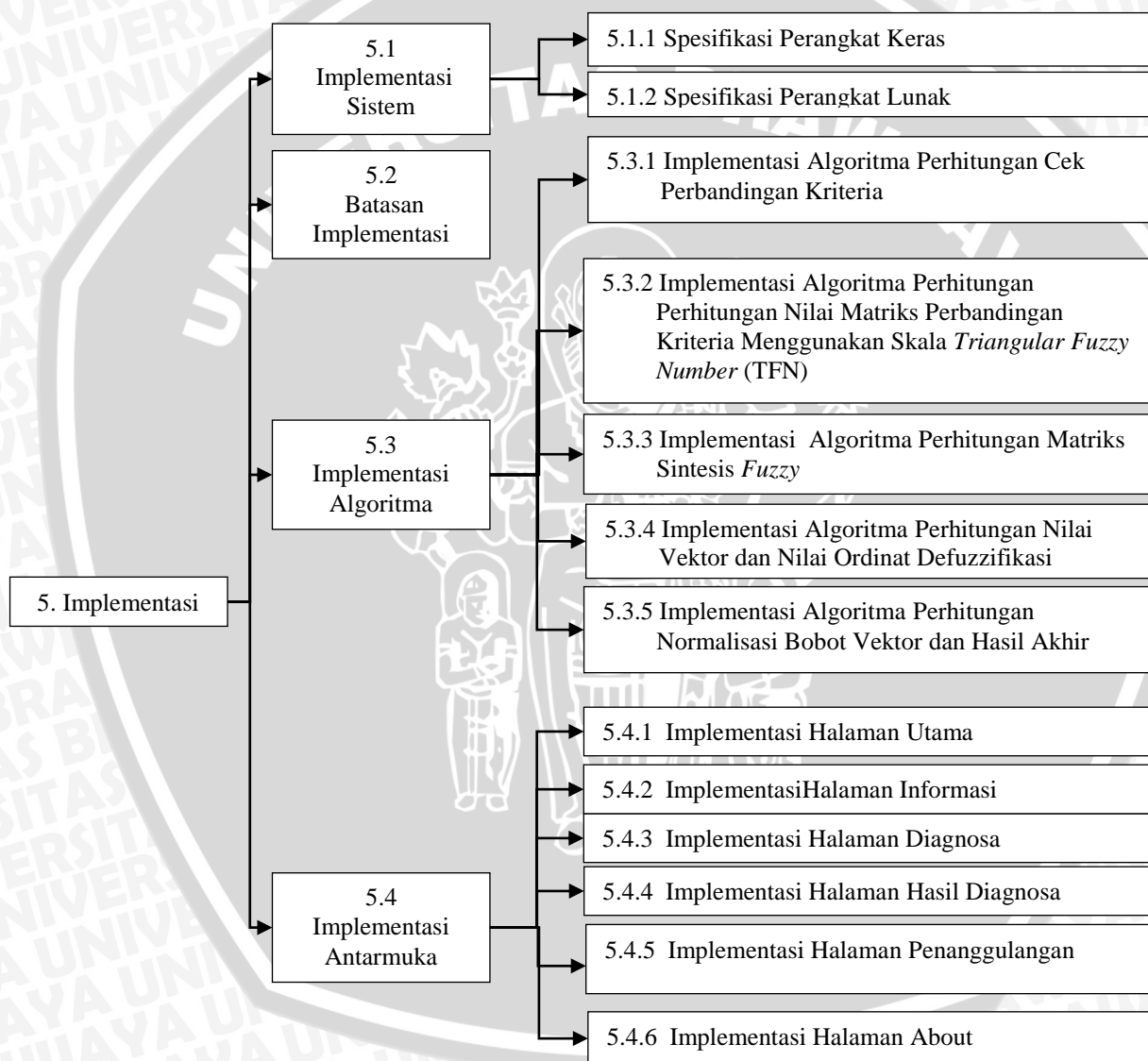


## BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi perangkat lunak, pembahasan terdiri dari penjelasan spesifikasi sistem, batasan implementasi, implementasi, algoritma, dan implementasi antarmuka. Secara garis besar bab implementasi ditunjukkan pada gambar 5.1.



**Gambar 5.1** Pohon Implementasi  
Sumber :[Implementasi]



## 5.1. Implementasi Sistem

Hasil dari proses analisa kebutuhan dan perancangan perangkat lunak yang telah dijelaskan pada Bab 4 menjadi acuan pada proses implementasi sistem. Spesifikasi perangkat yang dibutuhkan oleh sistem terdiri dari spesifikasi perangkat lunak dan spesifikasi perangkat keras.

### 5.1.1. Spesifikasi Perangkat Keras

Sistem pakar ini menggunakan perangkat komputer dengan spesifikasi perangkat keras seperti tabel 5.1.

**Tabel 5.1** Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel® Core™ i5-4200U CPU @ 1.60 GHz
Memori	4096 MB
Kartu Grafis	NVIDIA Geforce 720m
Harddisk	500 GB

Sumber : [Implementasi]

### 5.1.2. Spesifikasi Perangkat Lunak

Sistem pakar ini juga menggunakan perangkat komputer dengan spesifikasi perangkat lunak seperti pada tabel 5.2.

**Tabel 5.2** Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem operasi	Microsoft Windows 8 64-bit
Bahasa pemrograman	PHP
Tools Pemrograman	Netbeans IDE 7.3

Sumber : [Implementasi]

## 5.2. Batasan Implementasi

Berikut merupakan batasan implementasi pada sistem pakar diagnosa penyakit tanaman kopi *arabica* adalah sebagai berikut :

- a. Sistem dibangun berbasis web dengan menggunakan bahasa pemrograman *PHP*
- b. Metode yang digunakan untuk mendiagnosa penyakit menggunakan metode *Fuzzy AHP*
- c. Input dalam sistem pakar ini menggunakan parameter gejala dari kondisi tanaman, daun, batang, akar, dan buah
- d. Output berupa hasil perankingan dari perhitungan metode *Fuzzy AHP*
- e. Kriteria yang digunakan dalam penelitian sebanyak 5 kriteria.

### 5.3. Implementasi Algoritma

Pada implementasi algoritma menjelaskan tentang implementasi algoritma proses hitung metode fuzzy-AHP. Sistem pakar ini memiliki dua fungsi utama yaitu fungsi diagnosa penyakit tanaman kopi dan pengendalian penyakit berdasarkan hasil diagnosa.

#### 5.3.1. Implementasi Algoritma Proses Perhitungan Cek Konsistensi Perbandingan Kriteria

Pada proses ini dilakukan pengecekan konsistensi pada tabel perbandingan kriteria dengan aturan dikatakan kosisten jika nilai kurang dari 0.1. Pengecekan dilakukan dengan menggunakan metode AHP konvensional dengan mencari bobot kriteria untuk menentukan lamda maksimum.

##### a. Implementasi Algoritma Perhitungan Normalisasi Perbandingan Kriteria

Implementasi algoritma normalisasi terhadap matriks perbandingan berpasangan kriteria dilakukan dengan cara membagi tiap elemen matriks perbandingan dengan total tiap kolomnya seperti ditunjukkan pada gambar 5.2

1	//A1A1 Rule
2	\$Ruled16=1;
3	//A1A2 Rule
4	\$Rulee16=1/3;
5	//A1A3 Rule
6	\$Rulef16=1/5;
7	//A1A4 Rule
8	\$Ruleg16=2;
9	//A1A5
9	\$Ruleh16=1;
10	//A2A1 Rule
11	\$Ruled17=3;
12	//A2A2 Rule
13	\$Rulee17=1;
14	//A2A3 Rule
15	\$Rulef17=1/3;
16	//A2A4 Rule
17	\$Ruleg17=3;
18	//A2A5 Rule
19	\$Ruleh17=3;
20	//A3A1 Rule
21	\$Ruled18=5;
22	//A3A2 Rule
23	\$Rulee18=3;
24	//A3A3 Rule
25	\$Rulef18=1;
26	//A3A4 Rule
27	\$Ruleg18=7;
28	//A3A5 Rule
29	\$Ruleh18=5;
30	//A4A1 Rule
30	\$Ruled19=1/2;
31	//A4A2 Rule
	\$Rulee19=1/3;





```

32 //A4A3 Rule
33 $Rulef19=1/7;
34 //A4A4 Rule
35 $Ruleg19=1;
36 //A4A5
37 $Ruleh19=1/2;
38 //A5A1
39 $Ruled20=1;
40 //A5A2
41 $Rulee20=1/3;
42 //A5A3
43 $Rulef20=1/5;
44 //A5A4
45 $Ruleg20=2;
46 //A5A5
47 $Ruleh20=1;
48
49 //Jumlah Matriks Rule
50 $a1jumlahRule=$Ruled16+$Ruled17+$Ruled18+$Ruled19+$Ruled20;
51 $a2jumlahRule=$Rulee16+$Rulee17+$Rulee18+$Rulee19+$Rulee20;
52 $a3jumlahRule=$Rulef16+$Rulef17+$Rulef18+$Rulef19+$Rulef20;
53 $a4jumlahRule=$Ruleg16+$Ruleg17+$Ruleg18+$Ruleg19+$Ruleg20;
54 $a5jumlahRule=$Ruleh16+$Ruleh17+$Ruleh18+$Ruleh19+$Ruleh20;
55
56 //array Rule A1-A5
57 $a1ArrayRule= array($Ruled16, $Rulee16, $Rulef16, $Ruleg16, $Ruleh16);
58 $a2ArrayRule= array($Ruled17, $Rulee17, $Rulef17, $Ruleg17, $Ruleh17);
59 $a3ArrayRule= array($Ruled18, $Rulee18, $Rulef18, $Ruleg18, $Ruleh18);
60 $a4ArrayRule= array($Ruled19, $Rulee19, $Rulef19, $Ruleg19, $Ruleh19);
61 $a5ArrayRule= array($Ruled20, $Rulee20, $Rulef20, $Ruleg20, $Ruleh20);
62 //array matriks Rule
63 $arrayJumlahRule= array($a1jumlahRule, $a2jumlahRule, $a3jumlahRule,
64 $a4jumlahRule, $a5jumlahRule);
65 //Normalisasi Rule
66 for($i=0;$i<5;$i++){
67     $a1NormalisasiRule[$i]=$a1ArrayRule[$i]/$arrayJumlahRule[$i];
68     $a2NormalisasiRule[$i]=$a2ArrayRule[$i]/$arrayJumlahRule[$i];
69     $a3NormalisasiRule[$i]=$a3ArrayRule[$i]/$arrayJumlahRule[$i];
70     $a4NormalisasiRule[$i]=$a4ArrayRule[$i]/$arrayJumlahRule[$i];
71     $a5NormalisasiRule[$i]=$a5ArrayRule[$i]/$arrayJumlahRule[$i];
72 }

```

**Gambar 5.2.** Implementasi Algoritma Proses Penentuan Alternatif  
**Sumber :**[Implementasi]

Penjelasan algoritma proses seleksi alternatif obat perawatan kulit wajah pada Gambar 5.2 adalah sebagai berikut :

1. Pada baris ke-1 hingga ke-47 merupakan matriks perbandingan.
2. Melakukan normalisasi menggunakan rumus yang telah ditentukan, ditunjukkan pada baris ke-49 hingga baris ke-70

## b. Implementasi Algoritma Perhitungan Bobot Kriteria

Bobot kriteria diperoleh dengan cara mencari nilai rata-rata setiap baris pada matriks ternormalisasi. Pada gambar 5.3 merupakan implementasi dari algoritma perhitungan bobot kriteria.

```

1 //bobot alternatif Rule
2
3 $a1BobotAlternatifRule = array_sum($a1NormalisasiRule)/5;
4 $a2BobotAlternatifRule = array_sum($a2NormalisasiRule)/5;
5 $a3BobotAlternatifRule = array_sum($a3NormalisasiRule)/5;
6 $a4BobotAlternatifRule = array_sum($a4NormalisasiRule)/5;
7 $a5BobotAlternatifRule = array_sum($a5NormalisasiRule)/5;

```

**Gambar 5.3** Implementasi Algoritma Perhitungan Bobot Kriteria

**Sumber:**[Implementasi]

Pada baris ke-3 sampai ke-7 merupakan perhitungan bobot kriteria yaitu dengan penjumlahan setiap baris lalu dibagi dengan banyaknya kriteria yaitu 5.

## 2. Implementasi Algoritma Perhitungan Nilai Lamda Maksimum

Nilai lamda maksimum diperoleh dari perkalian matriks antara bobot kriteria dengan matriks perbandingan kriteria lalu hasil dari perkalian tersebut dibagi dengan bobot kriteria dan dicari nilai rata-rata dari hasil bagi tersebut yang ditunjukkan pada gambar 5.4.

```

1 $bobotAlternatifRule=array
2 ($a1BobotAlternatifRule,$a2BobotAlternatifRule,$a3BobotAlternatifR
3 ule,$a4BobotAlternatifRule,$a5BobotAlternatifRule);
4 //vektor jumlah bobot
5 $a1vektorBobotRule=0;
6 $a2vektorBobotRule=0;
7 $a3vektorBobotRule=0;
8 $a4vektorBobotRule=0;
9 $a5vektorBobotRule=0;
10
11 for($i=0;$i<5;$i++){
12 $a1vektorBobotRule+=$a1ArrayRule[$i]*$bobotAlternatifRule[$i];
13 $a2vektorBobotRule+=$a2ArrayRule[$i]*$bobotAlternatifRule[$i];
14 $a3vektorBobotRule+=$a3ArrayRule[$i]*$bobotAlternatifRule[$i];
15 $a4vektorBobotRule+=$a4ArrayRule[$i]*$bobotAlternatifRule[$i];
16 $a5vektorBobotRule+=$a5ArrayRule[$i]*$bobotAlternatifRule[$i];
17 }
18 // konsistensi
19 $AMax=(($a1vektorBobotRule/$a1BobotAlternatifRule)+($a2vektorBobot
20 Rule/$a2BobotAlternatifRule)+($a3vektorBobotRule/$a3BobotAlternati
21 fRule)+($a4vektorBobotRule/$a4BobotAlternatifRule)+($a5vektorBobot
22 Rule/$a5BobotAlternatifRule))/5;

```

**Gambar 5.4** Implementasi Algoritma Perhitungan Nilai Lamda Maksimum

**Sumber:**[Implementasi]



Penjelasan algoritma pada Gambar 5.4 adalah sebagai berikut:

1. Pada baris ke-11 hingga ke-16 merupakan perkalian matriks antara bobot kriteria dengan matriks perbandingan berpasangan
2. Pada baris ke-19 sampai dengan ke-22 merupakan perhitungan lamda maksimum

### 3. Implementasi Algoritma Perhitungan Nilai *Consistency Ratio* (CR)

Proses perhitungan nilai CR adalah dengan cara pembagian antara nilai *Consistency Index* (CI) dengan RI. Nilai CI diperoleh dengan cara lamda maksimum dikurangkan dengan jumlah kriteria lalu hasil dari pengurangan tersebut dibagi dengan jumlah kriteria dikurangkan 1. Untuk kriteria dengan jumlah 5 maka nilai RI adalah 1,12. Implementasi algoritma perhitungan nilai CR ditunjukkan pada gambar 5.5.

```

1  $ci=($AMax-5)/(5-1);
2  $konsistensiRule=$ci/1.12;
3
4  echo "$konsistensiRule";

```

**Gambar 5.5** Implementasi Algoritma Perhitungan Nilai *Consistency Ratio* (CR)  
**Sumber:**[Implementasi]

#### 5.3.2. Implementasi Algoritma Perhitungan Nilai Matriks Perbandingan Kriteria Menggunakan Skala *Triangular Fuzzy Number* (TFN)

Pada proses ini matriks perbandingan berpasangan kriteria ditransformasikan menggunakan skala TFN. Implementasi algoritma proses penentuan bobot kepentingan terdapat pada Gambar 5.6.

```

1  //skala TFN Rule
2  //A1
3
4  $arrayA1 = array();
5  $arrayA1[1] = $Ruled16; $arrayA1[4] = $Rulee16; $arrayA1[7] =
6  $Rulef16; $arrayA1[10] = $Ruleg16; $arrayA1[13] = $Ruleh16;
7
8  //a1
9  if ($arrayA1[1]==1) {$arrayA1[0]=1; $arrayA1[2]=3;}
10 if ($arrayA1[1]==2) {$arrayA1[0]=1; $arrayA1[2]=4;}
11 if ($arrayA1[1]==3) {$arrayA1[0]=1; $arrayA1[2]=5;}
12 if ($arrayA1[1]==4) {$arrayA1[0]=2; $arrayA1[2]=6;}
13 if ($arrayA1[1]==5) {$arrayA1[0]=3; $arrayA1[2]=7;}
14 if ($arrayA1[1]==6) {$arrayA1[0]=4; $arrayA1[2]=8;}
15 if ($arrayA1[1]==7) {$arrayA1[0]=5; $arrayA1[2]=9;}
16 if ($arrayA1[1]==8) {$arrayA1[0]=6; $arrayA1[2]=9;}
17 if ($arrayA1[1]==9) {$arrayA1[0]=7; $arrayA1[2]=9;}

```

```

18 //kebalikan
19 if ($arrayA1[1]==1/2) {$arrayA1[0]=1/4;$arrayA1[2]=1;}
20 if ($arrayA1[1]==1/3) {$arrayA1[0]=1/5;$arrayA1[2]=1;}
21 if ($arrayA1[1]==1/4) {$arrayA1[0]=1/6;$arrayA1[2]=1/2;}
22 if ($arrayA1[1]==1/5) {$arrayA1[0]=1/7;$arrayA1[2]=1/3;}
23 if ($arrayA1[1]==1/6) {$arrayA1[0]=1/8;$arrayA1[2]=1/4;}
24 if ($arrayA1[1]==1/7) {$arrayA1[0]=1/9;$arrayA1[2]=1/5;}
25 if ($arrayA1[1]==1/8) {$arrayA1[0]=1/9;$arrayA1[2]=1/6;}
26 if ($arrayA1[1]==1/9) {$arrayA1[0]=1/9;$arrayA1[2]=1/7;}

```

**Gambar 5.6** Implementasi Algoritma Perhitungan Nilai Matriks Perbandingan Kriteria Menggunakan Skala Triangular Fuzzy Number (TFN)

**Sumber** :[Implementasi]

Penjelasan algoritma proses penentuan bobot kepentingan pada Gambar

5.6 adalah sebagai berikut :

1. Pada baris ke-1 hingga pada baris ke-26 merupakan skala TFN pada baris alternatif pertama pada tabel 4.12.
2. Proses algoritma diatas hanya menjelaskan nilai middle.
3. Nilai middle berada pada array ke-1, ke-4, ke-7, ke-10 dan ke-13.
4. Untuk menentukan nilai lower dan upper ditentukan dari nilai middle.
5. Penjelasan ke-1 hingga ke-3 berlaku pada semua alternatif

### 5.3.3. Implementasi Algoritma Implementasi Algoritma Perhitungan Matriks Sintesis Fuzzy

Nilai sintesis *Fuzzy* dari setiap alternatif dihitung dengan cara jumlah nilai fuzzifikasi dalam satu baris untuk tiap variabel ( $l$ ,  $m$ , dan  $u$ ) dikalikan dengan invers dari jumlah keseluruhan tiap variabel ( $l$ ,  $m$ , dan  $u$ ). Jika matriks fuzzy =  $l$ ,  $m$ ,  $u$ , maka nilai dari fuzzy<sup>-1</sup> =  $1/u$ ,  $m$ ,  $l$ . Berikut algoritma perhitungan matriks sintesis *fuzzy* yang ditunjukkan pada gambar 5.7.

```

1 //lower
2 $jumlahlowerA1=$arrayA1[0]+$arrayA1[3]+$arrayA1[6]+$arrayA1[9]+$arrayA1[12];
3 $jumlahlowerA2=$arrayA2[0]+$arrayA2[3]+$arrayA2[6]+$arrayA2[9]+$arrayA2[12];
4 $jumlahlowerA3=$arrayA3[0]+$arrayA3[3]+$arrayA3[6]+$arrayA3[9]+$arrayA3[12];
5 $jumlahlowerA4=$arrayA4[0]+$arrayA4[3]+$arrayA4[6]+$arrayA4[9]+$arrayA4[12];
6 $jumlahlowerA5=$arrayA5[0]+$arrayA5[3]+$arrayA5[6]+$arrayA5[9]+$arrayA5[12];
7 //medium
8 $jumlahmediumA1=$arrayA1[1]+$arrayA1[4]+$arrayA1[7]+$arrayA1[10]+$arrayA1[13];
9 $jumlahmediumA2=$arrayA2[1]+$arrayA2[4]+$arrayA2[7]+$arrayA2[10]+$arrayA2[13];
10 $jumlahmediumA3=$arrayA3[1]+$arrayA3[4]+$arrayA3[7]+$arrayA3[10]+$arrayA3[13];
11 $jumlahmediumA4=$arrayA4[1]+$arrayA4[4]+$arrayA4[7]+$arrayA4[10]+$arrayA4[13];
12 $jumlahmediumA5=$arrayA5[1]+$arrayA5[4]+$arrayA5[7]+$arrayA5[10]+$arrayA5[13];
13 //upper
14 $jumlahupperA1=$arrayA1[2]+$arrayA1[5]+$arrayA1[8]+$arrayA1[11]+$arrayA1[14];
15 $jumlahupperA2=$arrayA2[2]+$arrayA2[5]+$arrayA2[8]+$arrayA2[11]+$arrayA2[14];
16 $jumlahupperA3=$arrayA3[2]+$arrayA3[5]+$arrayA3[8]+$arrayA3[11]+$arrayA3[14];
17 $jumlahupperA4=$arrayA4[2]+$arrayA4[5]+$arrayA4[8]+$arrayA4[11]+$arrayA4[14];
18 $jumlahupperA5=$arrayA5[2]+$arrayA5[5]+$arrayA5[8]+$arrayA5[11]+$arrayA5[14];

```



```

17
18 //jumlah total lower & medium & upper
19 $totalLower=$jumlahlowerA1+$jumlahlowerA2+$jumlahlowerA3+$jumlahlowerA4+$jumlahlowerA5;
20 $totalmedium=$jumlahmediumA1+$jumlahmediumA2+$jumlahmediumA3+$jumlahmediumA4+$jumlahmediumA5;
21 $totalupper=$jumlahupperA1+$jumlahupperA2+$jumlahupperA3+$jumlahupperA4+$jumlahupperA5;
22
23 // sintesis fuzzy
24
25 $sfA1Lower=$jumlahlowerA1/$totalupper;   $sfA1Medium=$jumlahmediumA1/$totalmedium;
26 $sfA1Upper=$jumlahupperA1/$totalLower;   $sfA2Medium=$jumlahmediumA2/$totalmedium;
27 $sfA2Lower=$jumlahlowerA2/$totalupper;   $sfA3Medium=$jumlahmediumA3/$totalmedium;
28 $sfA2Upper=$jumlahupperA2/$totalLower;   $sfA4Medium=$jumlahmediumA4/$totalmedium;
29 $sfA3Lower=$jumlahlowerA3/$totalupper;   $sfA5Medium=$jumlahmediumA5/$totalmedium;
30 $sfA3Upper=$jumlahupperA3/$totalLower;
31 $sfA4Lower=$jumlahlowerA4/$totalupper;   $sfA5Upper=$jumlahupperA5/$totalLower;
32 $sfA4Upper=$jumlahupperA4/$totalLower;
33 $sfA5Lower=$jumlahlowerA5/$totalupper;
34 $sfA5Upper=$jumlahupperA5/$totalLower;
35
36

```

**Gambar 5.7** Implementasi Algoritma Nilai Konsisten Menggunakan Perhitungan AHP.

**Sumber :**[Implementasi]

Penjelasan algoritma nilai konsisten menggunakan perhitungan AHP pada Gambar 5.7 adalah sebagai berikut :

1. Pada baris ke-1 hingga ke-17 merupakan cara mencari jumlah dari nilai  $l, m, u$  dari setiap alternatif.
2. Pada baris ke-19 sampai ke-24 merupakan cara mencari jumlah dari seluruh nilai  $l, m, u$ .
3. Pada baris ke-27 sampai ke-26 merupakan cara mencari nilai sintesis fuzzy.

### 5.3.4. Implementasi Algoritma Perhitungan Nilai Vektor dan Nilai Ordinat Defuzzifikasi

Setelah dilakukan perbandingan nilai sintesis fuzzy akan diperoleh nilai vektor dan ordinat defuzzifikasi ( $d'$ ) minimum. Berikut implementasi algoritma perhitungan nilai vektor dan nilai ordinat defuzzifikasi. Implementasi algoritma proses penentuan matriks keputusan terdapat pada Gambar 5.6.

```

1 //normalisasi bobot baris A1
2
3 if($sfA1Medium>=$sfA1Medium){$sintesisfuzzyA1[0]=1;}
4 elseif($sfA1Lower>=$sfA1Upper){$sintesisfuzzyA1[0]=0;}
5 else{$sintesisfuzzyA1[0]=($sfA1Lower-$sfA1Upper)/((($sfA1Medium-$sfA1Upper)-
6 ($sfA1Medium-$sfA1Lower));}
7
8 if($sfA2Medium>=$sfA1Medium){$sintesisfuzzyA1[1]=1;}
9 elseif($sfA1Lower>=$sfA2Upper){$sintesisfuzzyA1[1]=0;}
10 else{$sintesisfuzzyA1[1]=($sfA1Lower-$sfA2Upper)/((($sfA2Medium-$sfA2Upper)-
11 ($sfA1Medium-$sfA1Lower));}

```



```

12  if($sfA3Medium>=$sfA1Medium){$sintesisfuzzyA1[2]=1;}
13  elseif($sfA1Lower>=$sfA3Upper){$sintesisfuzzyA1[2]=0;}
14  else{$sintesisfuzzyA1[2]=($sfA1Lower-$sfA3Upper)/((($sfA3Medium-$sfA3Upper)-
15  ($sfA1Medium-$sfA1Lower));}
16
17  if($sfA4Medium>=$sfA1Medium){$sintesisfuzzyA1[3]=1;}
18  elseif($sfA1Lower>=$sfA4Upper){$sintesisfuzzyA1[3]=0;}
19  else{$sintesisfuzzyA1[3]=($sfA1Lower-$sfA4Upper)/((($sfA4Medium-$sfA4Upper)-
20  ($sfA1Medium-$sfA1Lower));}
21
22  if($sfA5Medium>=$sfA1Medium){$sintesisfuzzyA1[4]=1;}
23  elseif($sfA1Lower>=$sfA5Upper){$sintesisfuzzyA1[4]=0;}
24  else{$sintesisfuzzyA1[4]=($sfA1Lower-$sfA5Upper)/((($sfA5Medium-$sfA5Upper)-
25  ($sfA1Medium-$sfA1Lower));}
26
27  $minfuzzyA1=min($sintesisfuzzyA1[0],$sintesisfuzzyA2[0],$sintesisfuzzyA3[0],$sint
28  esisfuzzyA4[0],$sintesisfuzzyA5[0]);
29  $minfuzzyA2=min($sintesisfuzzyA1[1],$sintesisfuzzyA2[1],$sintesisfuzzyA3[1],$sint
30  esisfuzzyA4[1],$sintesisfuzzyA5[1]);
31  $minfuzzyA3=min($sintesisfuzzyA1[2],$sintesisfuzzyA2[2],$sintesisfuzzyA3[2],$sint
32  esisfuzzyA4[2],$sintesisfuzzyA5[2]);
33  $minfuzzyA4=min($sintesisfuzzyA1[3],$sintesisfuzzyA2[3],$sintesisfuzzyA3[3],$sint
34  esisfuzzyA4[3],$sintesisfuzzyA5[3]);
35  $minfuzzyA5=min($sintesisfuzzyA1[4],$sintesisfuzzyA2[4],$sintesisfuzzyA3[4],$sint
36  esisfuzzyA4[4],$sintesisfuzzyA5[4]);

```

**Gambar 5.8** Implementasi Algoritma Perhitungan Nilai Vektor dan Nilai Ordinat Defuzzifikasi

**Sumber :**[Implementasi]

Penjelasan algoritma proses penentuan bobot kepentingan pada Gambar 5.8 adalah sebagai berikut :

1. Pada baris ke-1 hingga ke-24 merupakan algoritma mencari nilai vektordan nilai ordinat pada kriteria pertama.
2. Cek jika nilai *middle* pada kriteria kedua lebih besar atau sama dengan nilai *middle* pada kriteria pertama maka nilai vektor adalah 1. Langkah ini ditunjukkan pada baris ke-3.
3. Cek jika nilai *lower* pada kriteria pertama lebih besar atau sama dengan nilai *upper* pada kriteria kedua maka nilai vektor adalah 0. Langkah ini ditunjukkan pada baris ke-4.
4. Jika tidak memenuhi kedua syarat tersebut di atas maka nilai vektor dihitung menggunakan rumus yang telah ditentukan. Langkah ini ditunjukkan pada baris ke-5 dan ke-6.
5. Proses diulang hingga kesemua kriteria.
6. Mencari nilai defuzzifikasi ditunjukkan pada baris ke-26 hingga baris ke-34.

### 5.3.5. Implementasi Algoritma Perhitungan Normalisasi Bobot Vektor dan Hasil Akhir

Normalisasi nilai bobot vektor diperoleh dari menghitung tiap elemen bobot vektor dibagi dengan jumlah bobot vektor itu sendiri. Jumlah bobot normalisasi adalah bernilai 1. Implementasi algoritma proses normalisasi matriks keputusan terdapat pada Gambar 5.9.

```

1  $finalbobotA1Rule=$minfuzzyA1/($minfuzzyA1+$minfuzzyA2+$minfuzzyA3+$minfuzzyA4+$m
2  infuzzyA5);
3  $finalbobotA2Rule=$minfuzzyA2/($minfuzzyA1+$minfuzzyA2+$minfuzzyA3+$minfuzzyA4+$m
4  infuzzyA5);
5  $finalbobotA3Rule=$minfuzzyA3/($minfuzzyA1+$minfuzzyA2+$minfuzzyA3+$minfuzzyA4+$m
6  infuzzyA5);
7  $finalbobotA4Rule=$minfuzzyA4/($minfuzzyA1+$minfuzzyA2+$minfuzzyA3+$minfuzzyA4+$m
8  infuzzyA5);
9  $finalbobotA5Rule=$minfuzzyA5/($minfuzzyA1+$minfuzzyA2+$minfuzzyA3+$minfuzzyA4+$m
10 infuzzyA5);
11 //bobot kriteria penyakit A1-A5
12
13 $finalbobotA1Tanaman; $finalbobotA1Batang; $finalbobotA1Daun; $finalbobotA1Akar;
14 $finalbobotA1Buah;          $finalbobotA1Rule;//tanaman
15
16 $finalbobotA2Tanaman; $finalbobotA2Batang; $finalbobotA2Daun; $finalbobotA2Akar;
17 $finalbobotA2Buah;          $finalbobotA2Rule;//batang
18
19 $finalbobotA3Tanaman; $finalbobotA3Batang; $finalbobotA3Daun; $finalbobotA3Akar;
20 $finalbobotA3Buah;          $finalbobotA3Rule;//daun
21
22 $finalbobotA4Tanaman; $finalbobotA4Batang; $finalbobotA4Daun; $finalbobotA4Akar;
23 $finalbobotA4Buah;          $finalbobotA4Rule;//akar
24
25 $finalbobotA5Tanaman; $finalbobotA5Batang; $finalbobotA5Daun; $finalbobotA5Akar;
26 $finalbobotA5Buah;          $finalbobotA5Rule;//buah
27
28 $hasilakhirA1=($finalbobotA1Tanaman*$finalbobotA1Rule)+($finalbobotA1Batang*$fina
29 lbobotA2Rule)+($finalbobotA1Daun*$finalbobotA3Rule)+($finalbobotA1Akar*$finalbobo
30 tA4Rule)+($finalbobotA1Buah*$finalbobotA5Rule);
31
32 $hasilakhirA2=($finalbobotA2Tanaman*$finalbobotA1Rule)+($finalbobotA2Batang*$fina
33 lbobotA2Rule)+($finalbobotA2Daun*$finalbobotA3Rule)+($finalbobotA2Akar*$finalbobo
34 tA4Rule)+($finalbobotA2Buah*$finalbobotA5Rule);
35
36 $hasilakhirA3=($finalbobotA3Tanaman*$finalbobotA1Rule)+($finalbobotA3Batang*$fina
37 lbobotA2Rule)+($finalbobotA3Daun*$finalbobotA3Rule)+($finalbobotA3Akar*$finalbobo
38 tA4Rule)+($finalbobotA3Buah*$finalbobotA5Rule);
39
40 $hasilakhirA4=($finalbobotA4Tanaman*$finalbobotA1Rule)+($finalbobotA4Batang*$fina
41 lbobotA2Rule)+($finalbobotA4Daun*$finalbobotA3Rule)+($finalbobotA4Akar*$finalbobo
42 tA4Rule)+($finalbobotA4Buah*$finalbobotA5Rule);
43
44 $hasilakhirA5=($finalbobotA5Tanaman*$finalbobotA1Rule)+($finalbobotA5Batang*$fina
45 lbobotA2Rule)+($finalbobotA5Daun*$finalbobotA3Rule)+($finalbobotA5Akar*$finalbobo
46 tA4Rule)+($finalbobotA5Buah*$finalbobotA5Rule);
47
48 $hasilIdentifikasi=max($hasilakhirA1,$hasilakhirA2,$hasilakhirA3,$hasilakhirA4,$h
49 asilakhirA5);

```



50	
51	
52	if(\$hasilIdentifikasi==\$hasilakhirA1){echo"<b>Tanaman kopi anda terkena penyakit Nematoda</b> ";}
53	
54	elseif(\$hasilIdentifikasi==\$hasilakhirA2){echo"<b>Tanaman kopi anda terkena penyakit Karat Daun</b>";}
55	
56	elseif(\$hasilIdentifikasi==\$hasilakhirA3){echo"<b>Tanaman kopi anda terkenapenyakit Bercak Daun</b>";}
57	
58	
59	elseif(\$hasilIdentifikasi==\$hasilakhirA4){echo"<b>Tanaman kopi anda terkena penyakit Antraknos</b>";}
60	
61	
62	elseif(\$hasilIdentifikasi==\$hasilakhirA5){echo"<b>Tanaman kopi anda terkena penyakit Jamur Upas</b> ";}

**Gambar 5.9** Implementasi Algoritma Perhitungan Normalisasi Bobot Vektor

**Sumber :** [Implementasi]

Penjelasan algoritma pada Gambar 5.7 adalah sebagai berikut:

1. Untuk menghitung sintesis fuzzy dengan menggunakan algoritma pada baris ke-1 hingga ke-10
2. Untuk menghitung hasil akhir yaitu dengan menggunakan algoritma pada baris ke-28 hingga ke-49.
3. Hasil akhir didapat dengan cara dipilih nilai tertinggi dari perkalian bobot kriteria setiap alternatif dengan bobot kriteria awal.
4. Pada baris ke-51 hingga ke-62 merupakan algoritma untuk menampilkan hasil akhir.

#### **5.4. Implementasi Antarmuka**

Antarmuka aplikasi sistem pakar ini digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Pada implementasi antarmuka terdiri dari implementasi antarmuka halaman utama, halaman diagnosa, halaman about, halaman informasi penyakit, halaman hasil diagnosa, dan halaman penanggulangan.

##### **5.4.1. Implementasi Antarmuka Halaman Utama**

Halaman utama merupakan halaman awal aplikasi sistem pakar ini. Halaman ini terdiri dari tiga tombol utama yaitu diagnosa, informasi, dan about. Sistem ini tidak menggunakan autentifikasi untuk menggunakannya jadi pengguna langsung bisa mengakses aplikasi sistem pakar ini dan langsung menggunakannya. Implementasi antarmuka halaman utamaterdapat pada Gambar 5.10.



**Gambar 5.10** Implementasi Antarmuka Halaman Utama  
Sumber : [Implementasi]

#### 5.4.2 .Implementasi Antarmuka Halaman Informasi

Halaman informasi merupakan halaman yang berisi tentang informasi penyakit pada tanaman kopi. Pada halaman ini terdapat tombol navigasi untuk setiap penyakit. Implementasi antarmuka halaman informasi terdapat pada Gambar 5.11.



**Gambar 5.11** Implementasi Antarmuka Halaman Informasi  
Sumber : [Implementasi]

#### 5.4.3 Implementasi Antarmuka Halaman Diagnosa

Halaman diagnosa merupakan halaman yang berisi mengenai pertanyaan-pertanyaan yang diajukan seputar gejala pada tanaman kopi *Arabica*. Pertanyaan ini yang nantinya digunakan sebagai pertimbangan diagnose penyakit pada tanaman kopi. Implementasi antarmuka halaman utama pakar terdapat pada gambar 5.12







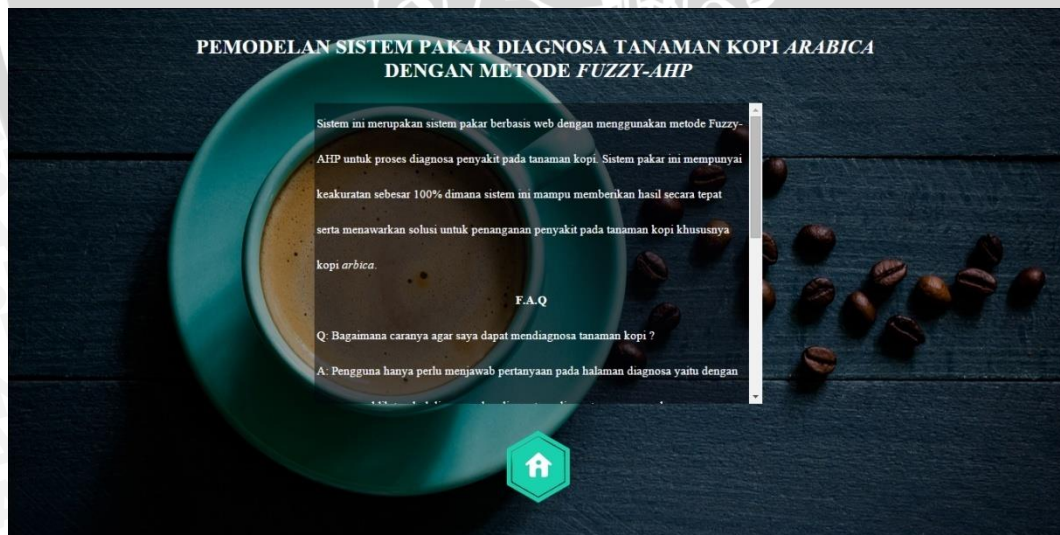
berhasil didiagnosa. Implementasi antarmuka halaman tambah atau ubah data pengguna terdapat pada Gambar 5.14



**Gambar 5.14** Implementasi Antarmuka Halaman Penanggulangan  
**Sumber :** [Implementasi]

#### 5.4.6 Implementasi Antarmuka Halaman About

Halaman about merupakan halaman yang berisi tentang aplikasi sistem pakar ini dan berisi tentang petunjuk penggunaan aplikasi ini. Halaman ini diakses melalui halaman utama. Implementasi antarmuka halaman pengguna terdapat pada Gambar 5.15



**Gambar 5.15** Implementasi Antarmuka Halaman About  
**Sumber :** [Implementasi]



<b>BAB V</b> .....	80
<b>IMPLEMENTASI</b> .....	80
<b>5.1. Implementasi Sistem</b> .....	81
<b>5.1.1. Spesifikasi Perangkat Keras</b> .....	81
<b>5.1.2. Spesifikasi Perangkat Lunak</b> .....	81
<b>5.2. Batasan Implementasi</b> .....	81
<b>5.3. Implementasi Algoritma</b> .....	82
<b>5.3.1. Implementasi Algoritma Proses Perhitungan Cek Konsistensi Perbandingan Kriteria</b> .....	82
<b>5.3.2. Implementasi Algoritma Perhitungan Nilai Matriks Perbandingan Kriteria Menggunakan Skala <i>Triangular Fuzzy Number</i> (TFN)</b> 85	
<b>5.3.3. Implementasi Algoritma Implementasi Algoritma Perhitungan Matriks Sintesis <i>Fuzzy</i></b> .....	86
<b>5.3.4. Implementasi Algoritma Perhitungan Nilai Vektor dan Nilai Ordinat Defuzzifikasi</b> .....	87
<b>5.3.5. Implementasi Algoritma Perhitungan Normalisasi Bobot Vektor dan Hasil Akhir</b> .....	89
<b>5.4. Implementasi Antarmuka</b> .....	90
<b>5.4.1. Implementasi Antarmuka Halaman Utama</b> .....	90
<b>5.4.2. Implementasi Antarmuka Halaman Informasi</b> .....	91
<b>5.4.3 Implementasi Antarmuka Halaman Diagnosa</b> .....	91
<b>5.4.4 Implementasi Antarmuka Halaman Hasil Diagnosa</b> .....	92
<b>5.4.5 Implementasi Antarmuka Halaman Penanggulangan</b> .....	92
<b>5.4.6 Implementasi Antarmuka Halaman About</b> .....	93