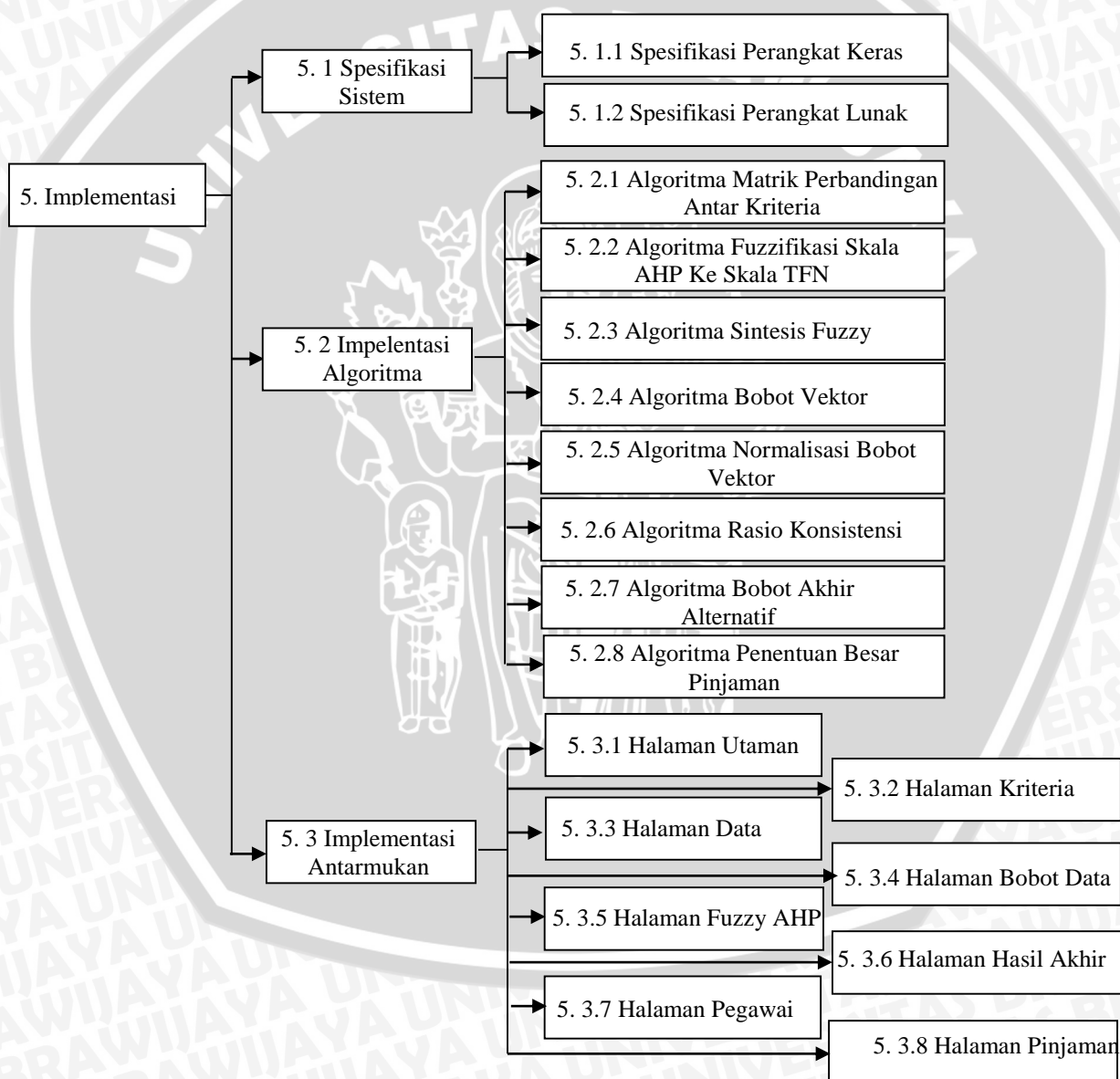


## BAB V IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma pada program, implementasi antarmuka, dan implementasi metode.



**Gambar 5.1** Pohon Implementasi  
Sumber : implementasi



## 5.1. Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

### 5.1.1. Spesifikasi Perangkat Keras

Pengembangan Sistem Pendukung Keputusan penentuan besar pinjaman pada koperasi menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core(TM) Duo CPU T5870 @ 2.00GHz
Memory (RAM)	2048 MB
Harddisk	160 GB

Sumber : Implementasi

### 5.1.2. Spesifikasi Perangkat Lunak

Pengembangan Sistem Pendukung Keputusan penentuan besar pinjaman pada koperasi menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem Operasi	Microsoft Windows 7 Ultimate 32-bit
Bahasa Pemrograman	Java
Tool Pemrograman	Netbeans 7.1
Server Localhost	XAMPP Server Version 1.7.3
DBMS	MySQL
Tool DBMS	MySQL Version 5.1.41

Sumber : Implementasi

## 5.2. Implementasi Algoritma

Sistem Pendukung Keputusan penentuan besar pinjaman pada koperasi menggunakan metode Fuzzy AHP memiliki beberapa algoritma yaitu algoritma matrik perbandingan antar kriteria, algoritma fuzzyfikasi skala AHP ke skala TFN,

algoritma sintesis fuzzy, algoritma bobot vektor, algoritma normalisasi bobot vektor, algoritma rasio konsistensi, algoritma bobot akhir alternatif dan algoritma penentuan besar pinjaman.

### 5.2.1. Implementasi Algoritma Matrik Perbandingan Antar kriteria

Prose pembuatan matrik perbandingan antar kriteria dilakukan dengan cara mengambil data nilai kriteria dari database. Apabila data akan diperbarui maka user menginputkan data yang baru ke sistem dan secara otomatis akan disimpan dalam database. Gambar 5.2. merupakan implementasi algoritma matrik perbandingan antar kriteria yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian B.

```

1 public void bobot_kriteria(){
2     double[][] data=new double[4][4];
3     for (int i = 0; i < 4; i++) {
4         for (int j = 0; j < 4; j++) {
5             data[i][j]=Double.parseDouble(data_kriteria.getValueAt(i,
6             j+1).toString().replace(",","."));
7         }
8     }
9 }

```

**Gambar 5.2** Implementasi Algoritma Matrik Perbandingan Antar Kriteria

Sumber : implementasi

Penjelasan algoritma matrik perbandingan antar kriteria pada gambar 5.2 yaitu :

1. Baris 2 menjelaskan pembuatan matrik 2 dimensi dengan dengan panjang 4
2. Baris 3-8 menjelaskan pengambilan data kriteria berpasangan dari tabel interface yang kemudian disimpan dalam array data

### 5.2.2. Implementasi Algoritma Fuzzyfikasi Skala AHP ke Skala TFN

Proses fuzzyfikasi skala AHP ke skala TFN dilakukan dengan cara mentransofmasikan matrik perbandingan kedalam skala Fuzzy berdasarkan tabel 2.4.. Gambar 5.3. merupakan implementasi algoritma fuzzyfikasi skala AHP ke skala TFN yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian C.

```

1 public void banding_pasangan(){
2     for (int i = 0; i < m.length; i++) {
3         for (int j = 0; j < m.length; j++) {
4             if(i==j) {
5                 l[i][j]=pembobotan_l(m[i][j],true,true);
6                 u[i][j]=pembobotan_u(m[i][j],true,true);
7             }
8             else {

```

```
9         if(i<j){
10             l[i][j]=pembobotan_l(m[i][j], false, true);
11             u[i][j]=pembobotan_u(m[i][j], false, true);
12         }
13         else{
14             l[i][j]=pembobotan_l(m[i][j],false,false);
15             u[i][j]=pembobotan_u(m[i][j],false,false);
16         }
17     }
18 }
19 }
20 }
21
22 public double pembobotan_l(double data,boolean
23 diagonal,boolean atas){
24     double hasil=0;
25     if(data==1 && diagonal)hasil=1;
26     else if(data==1 && !diagonal){
27         if(!atas)
28             hasil=0.333;
29         else hasil=1;
30     }
31     else if(data==2)hasil=1;
32     else if(data==3)hasil=1;
33     else if(data==4)hasil=2;
34     else if(data==5)hasil=3;
35     else if(data==6)hasil=4;
36     else if(data==7)hasil=5;
37     else if(data==8)hasil=6;
38     else if(data==9)hasil=7;
39
40     else if(data==0.5)hasil=0.25;
41     else if(data==0.333)hasil=0.2;
42     else if(data==0.25)hasil=0.167;
43     else if(data==0.2)hasil=0.143;
44     else if(data==0.167)hasil=0.125;
45     else if(data==0.143)hasil=0.111;
46     else if(data==0.125)hasil=0.111;
47     else if(data==0.111)hasil=0.111;
48     return hasil;
49 }
50
51 public double pembobotan_u(double data,boolean
52 diagonal,boolean atas){
53     double hasil=0;
54     if(data==1 && diagonal)hasil=1;
55     else if(data==1 && !diagonal){
56         if(!atas)
57             hasil=1;
58         else hasil=3;
59     }
60     else if(data==2)hasil=4;
61     else if(data==3)hasil=5;
62     else if(data==4)hasil=6;
63     else if(data==5)hasil=7;
64     else if(data==6)hasil=8;
65     else if(data==7)hasil=9;
```

```

66     else if(data==8)hasil=9;
67     else if(data==9)hasil=9;
68
69     else if(data==0.5)hasil=1;
70     else if(data==0.333)hasil=1;
71     else if(data==0.25)hasil=0.5;
72     else if(data==0.2)hasil=0.333;
73     else if(data==0.167)hasil=0.25;
74     else if(data==0.143)hasil=0.2;
75     else if(data==0.125)hasil=0.167;
76     else if(data==0.111)hasil=0.143;
77     return hasil;
78 }

```

**Gambar 5.3** Implementasi Algoritma Fuzzyfikasi Skala AHP ke Skala TFN

Sumber : implementasi

Penjelasan algoritma fuzzyfikasi skala AHP ke skala TFN pada gambar 5.3 yaitu :

1. Baris 2-3 menjelaskan perulangan untuk membaca matrik perbandingan berpasangan.
2. Baris 4-7 menjelaskan kondisi untuk mengecek apakah letak matrik  $m[i][i]$  merupakan diagonal.
3. Baris 9-16 menjelaskan kondisi apakah letak matrik  $m$  diatas diagonal atau dibawah diagonal.
4. Baris 22-49 menjelaskan penentuan nilai  $l$  pada skala TFN berdasarkan nilai matrik  $m$ .
5. Baris 51-78 menjelaskan penentuan nilai  $m$  pada skala TFn berdasarkan nilai matrik  $m$ .

### 5.2.3. Implementasi Algoritma Sintesis Fuzzy

Proses perhitungan nilai sisntesis fuzzy dilakukan dengan cara menjumlahkan nilai  $l$ ,  $m$ , dan  $u$  tiap kriteria dan hitung seluruh nila total  $l$ ,  $m$ , dan  $u$ . Kemudian bagi nilai  $l$ ,  $m$ , dan  $u$  tiap kriteria dengan total  $l$ ,  $m$ , dan  $u$ . Gambar 5.4. merupakan implementasi algoritma sintesis fuzzy yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian D.

```

1 private void sintesis(){
2     for (int i = 0; i < m.length; i++) {
3         double sum_m=0,sum_l=0,sum_u=0;
4         for (int j = 0; j < m.length; j++) {
5             sum_m+=m[i][j];
6             sum_l+=l[i][j];
7             sum_u+=u[i][j];
8         }
9         jumlah m[i]=sum m;

```

```

10     jumlah_l[i]=sum_l;
11     jumlah_u[i]=sum_u;
12     }
13     for (int i = 0; i < m.length; i++) {
14         total_m+=jumlah_m[i];
15         total_l+=jumlah_l[i];
16         total_u+=jumlah_u[i];
17     }
18     for (int i = 0; i < m.length; i++) {
19         sintesis[i][0]=jumlah_l[i]/total_u;
20         sintesis[i][1]=jumlah_m[i]/total_m;
21         sintesis[i][2]=jumlah_u[i]/total_l;
22     }
23     }

```

**Gambar 5.4** Implementasi Algoritma Sintesis Fuzzy

Sumber : implementasi

Penjelasan algoritma sintesis fuzzy pada gambar 5.4 yaitu :

1. Baris 2-12 menjelaskan perulangan untuk menghitung jumlah  $l$ ,  $m$  dan  $u$  dalam 1 baris.
2. Baris 13-17 menjelaskan perulangan untuk menghitung nilai total seluruh  $l$ ,  $m$  dan  $u$ .
3. Baris 18-23 menjelaskan perulangan untuk menghitung nilai sistesis.

#### 5.2.4. Implementasi Algoritma Bobot Vektor

Proses perhitungan nilai bobot vektor dilakukan dengan data sintesis fuzzy yang kemudian diolah dengan persamaan (2-14) dan (2-15). Gambar 5.5. merupakan implementasi algoritma sintesis fuzzy yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian E.

```

1     public void nilaiVektor(){
2         for (int i = 0; i < m.length; i++) {
3             for (int j = 0; j < m.length; j++) {
4                 if(i!=j){
5                     if(sintesis[i][1]>=sintesis[j][1])nilai_vektor[i][j]=1;
6                     else
7                     if(sintesis[j][0]>=sintesis[i][2])nilai_vektor[i][j]=0;
8                     else {
9                         nilai_vektor[i][j]=(sintesis[j][0]-
10                        sintesis[i][2])/((sintesis[i][1]-sintesis[i][2])-
11                        (sintesis[j][1]-sintesis[j][0]));
12                     }
13                 }
14             }
15         }
16     }
17
18     private void bobotVektor(){
19         minimal=new double[m.length];

```

```

20     for (int i = 0; i < m.length; i++) {
21         minimal[i]=9999;
22         for (int j = 0; j < m.length; j++) {
23             if(i!=j)
24                 minimal[i]=Math.min(minimal[i], nilai_vektor[i][j]);
25         }
26     }
27 }

```

**Gambar 5.5** Implementasi Algoritma Bobot Vektor

Sumber : implementasi

Penjelasan algoritma bobot vektor pada gambar 5.5 yaitu :

1. Baris 2-16 menjelaskan perulangan untuk mencari nilai vektor.
2. Baris 18-27 menjelaskan proses pencarian nilai bobot vektor berdasarkan nilai vektor.

### 5.2.5. Implementasi Algoritma Normalisasi Bobot Vektor

Proses perhitungan normalisasi bobot vektor dilakukan dengan cara membagi nilai setiap bobot vektor dengan nilai jumlah seluruh bobot vektor.

Gambar 5.6. merupakan implementasi algoritma sintesis fuzzy yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian F.

```

1     private void normalisasiBobotVektor() {
2         double sum=0;
3         for (int i = 0; i < m.length; i++) {
4             sum+=minimal[i];
5         }
6         for (int i = 0; i < m.length; i++) {
7             bobot_vektor[i]=minimal[i]/sum;
8         }
9     }
10 }

```

**Gambar 5.6** Implementasi Algoritma Normalisasi Bobot Vektor

Sumber : implementasi

Penjelasan algoritma normalisasi bobot vektor pada gambar 5.6 yaitu :

1. Baris 3-6 menjelaskan perulangan untuk menghitung jumlah nilai bobot vektor.
2. Baris 7-8 menjelaskan perulangan untuk mencari nilai normalisasi bobot vektor.

### 5.2.6. Implementasi Algoritma Rasio Konsistensi

Proses perhitungan rasio konsistensi dijelaskan pada gambar 4.17., Gambar 5.7. merupakan implementasi algoritma rasio konsistensi yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian G.

```

1 public void consistencyRatio() {
2     double sum=0;
3     double[] r=new double[bobot_vektor.length];
4     for (int i = 0; i < bobot_vektor.length; i++) {
5         for (int j = 0; j < 3; j++) {
6             if(bobot_vektor[i]!=0)
7                 r[i]+=sintesis[i][j]/bobot_vektor[i];
8         }
9         r[i]/=3;
10    }
11    for (int i = 0; i < r.length; i++) {
12        sum+=r[i];
13    }
14    double
15    eigen_value=Math.round(sum*1e2)/1e2/bobot_vektor.length;
16    double
17    CI=(eigen_value-bobot_vektor.length)/bobot_vektor.length-1);
18    CR=CI/IR;
19 }

```

**Gambar 5.7** Implementasi Algoritma Rasio Konsistensi

Sumber : implementasi

Penjelasan algoritma rasio konsistensi pada gambar 5.7 yaitu :

1. Baris 4-10 menjelaskan perulangan untuk menghitung nilai sintesis/bobot vektor.
2. Baris 11-13 menjelaskan perulangan untuk mengitung jumlah hasil pembagian nilai sisntesis/bobot vektor.
3. Baris 15 menjelaskan perhitungan untuk menghitung nilai eigen value.
4. Baris 17 menjelaskan perhitungan untuk mencari nilai indeks konsistensi (CI).
5. Baris 18 menjelaskan perhitungan nilai rasio konsistensi (CR).

### 5.2.7. Implementasi Algoritma Bobot Akhir Alternatif

Proses perhitungan bobot akhir alternatif dilakukan dengan cara mengalikan nilai bobot global kriteria dengan nilai bobot global alternatif tiap kriteria, dan kemudian dilakukan proses penjumlahan. Gambar 5.8. merupakan implementasi algoritma bobot akhir alternatif yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian L.



```

1 HitungPinjaman(double[] kriteria,double[] umur,double[]
2 pekerjaan,double[] lapin,double[] gaji,int[] data_lapin){
3     bobot_akhir=new double[umur.length];
4     for (int i = 0; i < umur.length; i++) {
5         bobot_akhir[i]+=kriteria[0]*umur[i];
6         bobot_akhir[i]+=kriteria[1]*pekerjaan[i];
7         bobot_akhir[i]+=kriteria[2]*lapin[i];
8         bobot_akhir[i]+=kriteria[3]*gaji[i];
9     }
10 }

```

**Gambar 5.8** Implementasi Algoritma Bobot Akhir Alternatif

Sumber : implementasi

Penjelasan algoritma bobot akhir alternatif pada gambar 5.8 yaitu :

1. Baris 3 menjelaskan inisialisasi matriks bobot\_akhir
2. Baris 4-9 menjelaskan perhitungan nilai bobot\_akhir tiap alternatif

### 5.2.8. Implementasi Algoritma Penentuan Besar Pinjaman

Proses perhitungan penentuan besar pinjaman dilakukan dengan cara mengalikan nilai bobot akhir alternatif dengan 30 dan dikalikan lagi dengan lama pinjaman. Gambar 5.9. merupakan implementasi algoritma penentuan besar pinjaman yang mengimplementasikan perancangan Sub Bab 4.2.3 bagian M.

```

1 public void x30() {
2     for (int i = 0; i < x30.length; i++) {
3         x30[i]=Math.round(bobot_akhir[i]*1e3)/1e3*30;
4     }
5 }
6 public void besar_pinjaman(){
7     for (int i = 0; i < besar_pinjaman.length; i++) {
8         besar_pinjaman[i]=x30[i]*lapin[i];
9     }
10 }

```

**Gambar 5.9** Implementasi Algoritma Penentuan Besar Pinjaman

Sumber : implementasi

Penjelasan algoritma bobot akhir alternatif pada gambar 5.9 yaitu :

1. Baris 2-4 menjelaskan perulangan untuk menghitung nilai bobot akhir x 30.
2. Baris 7-9 menjelaskan perulangan untuk menghitung besar pinjaman maksimum yang dapat diberikan.

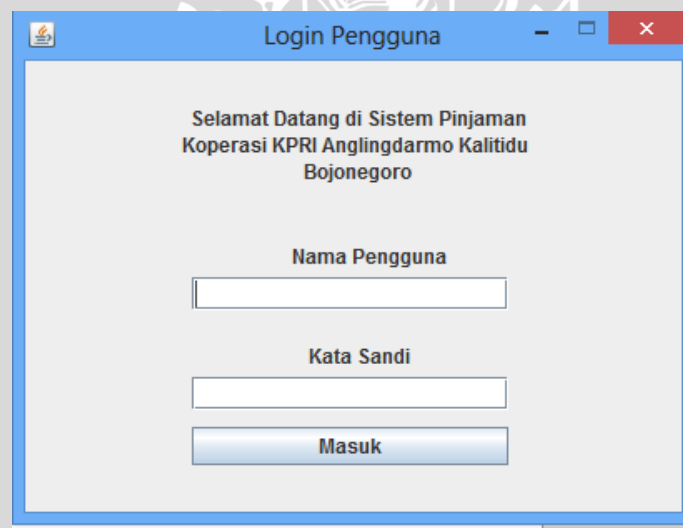
### 5.3 Implementasi Antar Muka

Antarmuka Sistem Pendukung Keputusan penentuan besar pinjaman digunakan oleh pengguna untuk berinteraksi dengan sistem. Antarmuka Sistem

Pendukung Keputusan antara lain antarmuka halaman utaman, Halaman kriteria, halaman data, halaman bobot data, halaman Fuzzy AHP, halaman hasil akhir, halaman pegawai dan halaman pinjaman.

### 5.3.1 Tampilan Halaman Utama

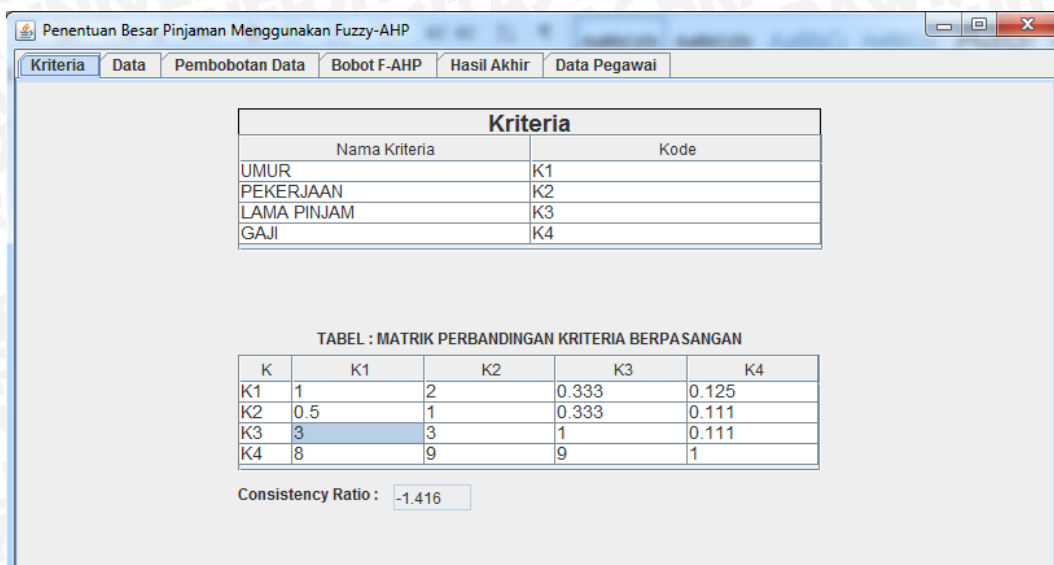
Halaman utama pada Sistem Pendukung Keputusan ini adalah halaman login. Halaman login digunakan untuk mengidentifikasi pengguna dalam sistem ini. Pengguna dalam sistem ini dibagi menjadi dua yaitu pimpinan dan pegawai. Pada halaman Login, pengguna dapat memulai login dengan memasukkan username dan password pada textbox yang tersedia. Tombol login berguna untuk memulai login setelah memasukkan username dan password. Gambar 5.10 merupakan implementasi antar muka halaman utama yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 1.



**Gambar 5.10** Implementasi Antarmuka Halaman Utama  
Sumber : implementasi

### 5.3.2 Tampilan Halaman Kriteria

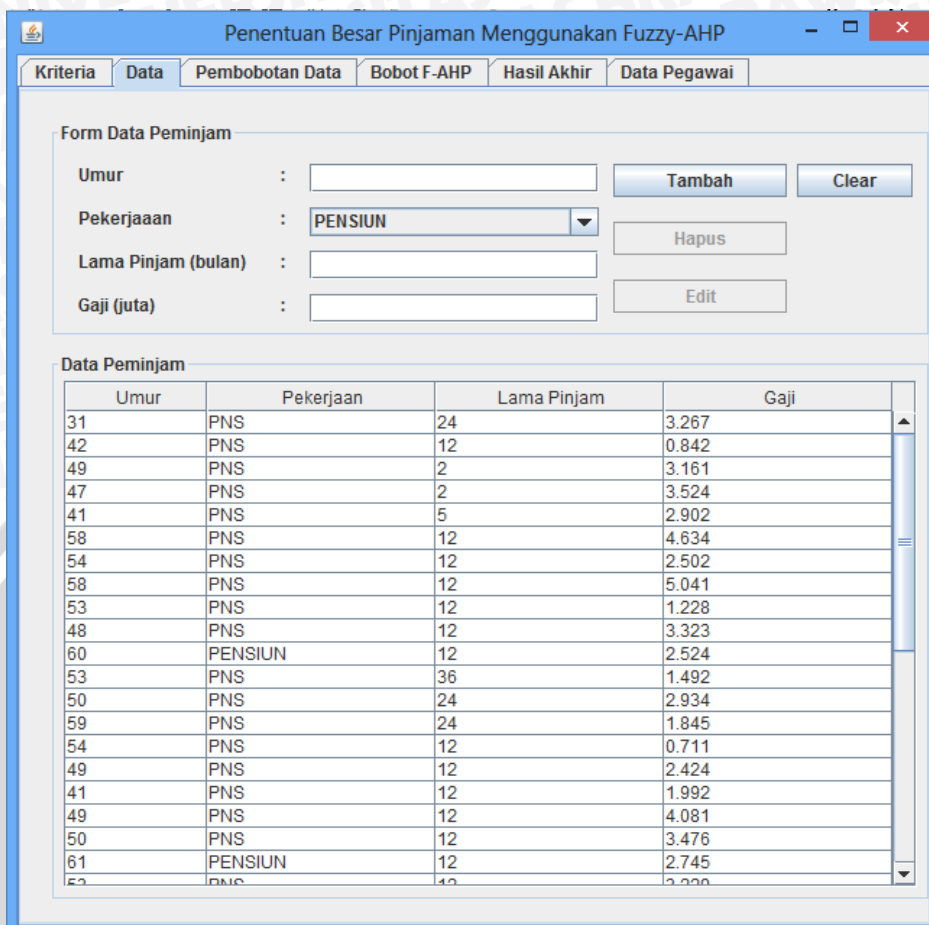
Halaman kriteria merupakan halaman yang digunakan pimpinan untuk melihat dan mengatur tingkat kepentingan dari kriteria. Pada halaman ini terdapat tabel perbandingan berpasangan dan terdapat proses pengecekan rasio konsistensi. Gambar 5.11 merupakan implementasi antar muka halaman kriteria yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 2.



**Gambar 5.11** Implementasi Antarmuka Halaman Kriteria  
 Sumber : implementasi

### 5.3.3 Tampilan Halaman Data

Halaman data merupakan halaman yang digunakan pengguna untuk melihat dan mengatur data calon kreditur. Pada halaman ini pengguna dapat memasukkan data calon kreditur yaitu umur, pekerjaan, lama pinjaman dan gaji pada textbox yang telah disediakan. Pada halaman ini terdapat tabel data yang menyimpan data calon kreditur. Gambar 5.12 merupakan implementasi antar muka halaman data yang mengimplemantasikan perancangan Sub Bab 4.2.4 bagian 3.



**Gambar 5.12** Implementasi Antarmuka Halaman Data  
 Sumber : implementasi

**5.3.4 Tampilan Halaman Bobot Data**

Halaman bobot data merupakan halaman yang digunakan pengguna untuk melihat tabel aturan pembobotan data calon kreditur dan data hasil pembobotan. Gambar 5.13 merupakan implementasi antar muka halaman bobot data yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 4.

repository.ub.ac.id

Keterangan Bobot Umur	
Umur	Bobot
Umur <= 35	6
35 < Umur <= 40	5
40 < Umur <= 45	4
45 < Umur <= 50	3
50 < Umur <= 55	2

Keterangan Bobot Pekerjaan	
Pekerjaan	Bobot
Pensiun	1
PNS	2

Keterangan Bobot Lama Pinjam	
Lapin	Bobot
Lapin <= 6	1
6 < Lapin <= 12	2
12 < Lapin <= 18	3
18 < Lapin <= 24	4
24 < Lapin <= 30	5

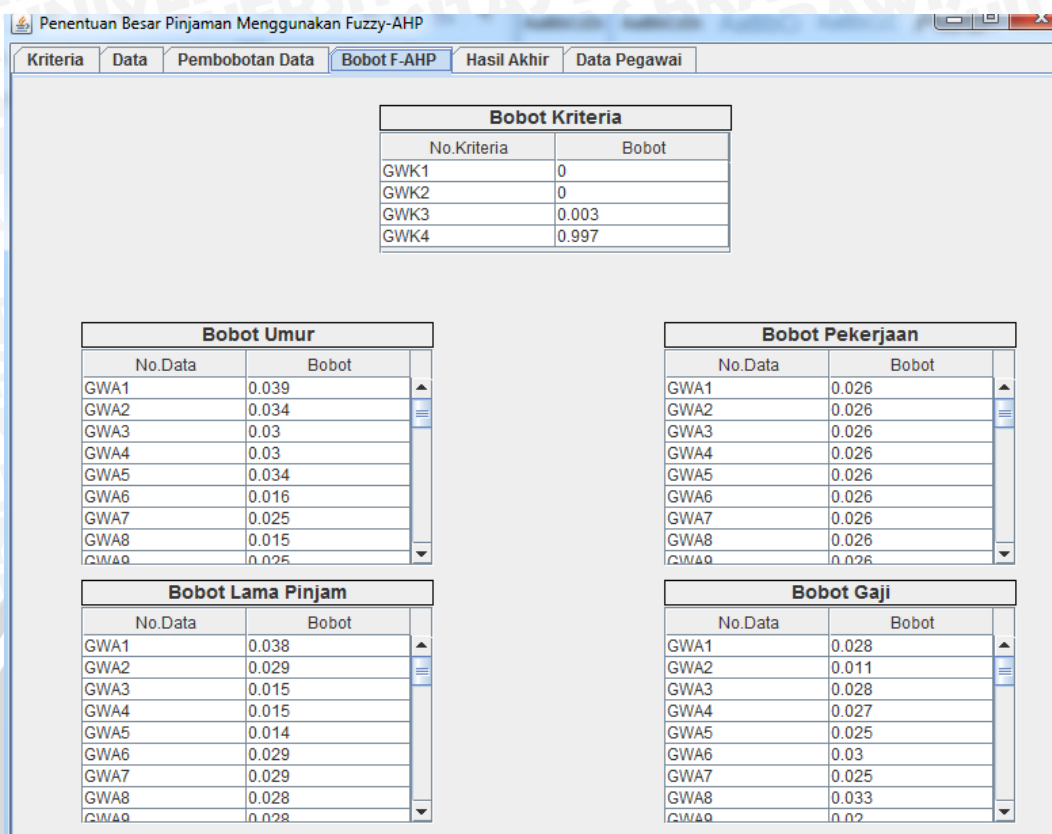
Keterangan Bobot Gaji	
Gaji	Bobot
Gaji <= 1	1
1 < Gaji <= 2	2
2 < Gaji <= 3	3
3 < Gaji <= 4	4
4 < Gaji <= 5	5

Hasil Pembobotan Data				
Umur	Pekerjaan	Lama Pinjam	Gaji	
6	2	4	4	
4	2	2	1	
3	2	1	4	
3	2	1	4	
4	2	1	3	
1	2	2	5	
2	2	2	3	
1	2	2	6	
2	2	2	2	
3	2	2	4	
1	1	2	3	
2	2	6	2	

**Gambar 5.13** Implementasi Antarmuka Halaman Pembobotan Data  
Sumber : implementasi

### 5.3.5 Tampilan Halaman Bobot F-AHP

Halaman bobot F-AHP merupakan halaman yang digunakan pengguna untuk melihat tabel hasil proses Fuzzy AHP yaitu bobot global kriteria dan bobot alternatif per kriteria. Gambar 5.14 merupakan implementasi antar muka halaman bobot F-AHP yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 5.



**Gambar 5.14** Implementasi Antarmuka Halaman Bobot F-AHP  
 Sumber : implementasi

### 5.3.6 Tampilan Halaman Hasil Akhir

Halaman hasil akhir merupakan halaman yang digunakan pengguna untuk melihat hasil bobot akhir alternatif dan hasil pinjaman maksimum yang dapat diberikan. Gambar 5.15 merupakan implementasi antar muka halaman hasil akhir yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 6.

Kriteria	Data	Pembobotan Data	Bobot F-AHP	Hasil Akhir	Data Pegawai
	Data	Bobot	x30	Lama Pinjaman	Pinjaman yang Diberikan (juta)
A1	0.028	0.028	0.84	24	20.16
A2	0.011	0.011	0.33	12	3.96
A3	0.027	0.027	0.81	2	1.62
A4	0.027	0.027	0.81	2	1.62
A5	0.025	0.025	0.75	5	3.75
A6	0.03	0.03	0.9	12	10.8
A7	0.025	0.025	0.75	12	9
A8	0.033	0.033	0.99	12	11.88
A9	0.02	0.02	0.6	12	7.2
A10	0.028	0.028	0.84	12	10.08
A11	0.025	0.025	0.75	12	9
A12	0.019	0.019	0.57	36	20.52
A13	0.024	0.024	0.72	24	17.28
A14	0.019	0.019	0.57	24	13.68
A15	0.01	0.01	0.3	12	3.6
A16	0.024	0.024	0.72	12	8.64
A17	0.019	0.019	0.57	12	6.84
A18	0.03	0.03	0.9	12	10.8
A19	0.028	0.028	0.84	12	10.08
A20	0.024	0.024	0.72	12	8.64
A21	0.028	0.028	0.84	12	10.08
A22	0.024	0.024	0.72	6	4.32
A23	0.03	0.03	0.9	2	1.8
A24	0.03	0.03	0.9	6	5.4
A25	0.03	0.03	0.9	36	32.4
A26	0.019	0.019	0.57	5	2.85
A27	0.028	0.028	0.84	5	4.2
A28	0.024	0.024	0.72	12	8.64
A29	0.03	0.03	0.9	12	10.8
A30	0.024	0.024	0.72	12	8.64
A31	0.023	0.023	0.69	12	8.28
A32	0.031	0.031	0.93	12	11.16
A33	0.028	0.028	0.84	3	2.52
A34	0.023	0.023	0.69	12	8.28

**Gambar 5.15** Implementasi Antarmuka Halaman Hasil Akhir  
 Sumber : implementasi

### 5.3.7 Tampilan Halaman Data Pegawai

Halaman data pegawai merupakan halaman yang digunakan pimpinan untuk melihat mengatur pengguna dalam sistem ini. Pada tampilan ini pimpinan data menambahkan data pengguna, nemedit dan menghapus data pengguna dari sistem ini. Pimpinan dapat nambahnkan data pengguan sistem dengan cara mengimputkan data calon pengguna sistem pada textbox yang telah disediakan. Gambar 5.16 merupakan implementasi antar muka halaman akun yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 7.

The screenshot shows a web application window titled "Penentuan Besar Pinjaman Menggunakan Fuzzy-AHP". The window has several tabs: "Kriteria", "Data", "Pembobotan Data", "Bobot F-AHP", "Hasil Akhir", and "Data Pegawai". The "Data Pegawai" tab is active, displaying a form and a table.

**Form Data Pegawai**

Nama Pengguna :

Jabatan :

Kata Sandi :

**Data Pegawai**

ID	Nama Pengguna	Jabatan	Kata Sandi
2	beta	PEGAWAI	beta
3	admin	MANAJER	admin

**Gambar 5.16** Implementasi Antarmuka Halaman Data Pegawai  
Sumber : implementasi

### 5.3.8 Tampilan Halaman Pinjaman

Halaman pinjaman merupakan halaman yang digunakan pegawai untuk menghitung besar pinjaman maksimum yang dapat diberikan kepada calon kreditur. Pada tampilan ini pegawai dapat melakukan perhitungan dengan cara mengumpulkan data calon kreditur berupa umur, pekerjaan, lama pinjaman, dan gaji pada textbox yang telah disediakan. Gambar 5.17 merupakan implementasi antar muka halaman pinjaman yang mengimplemaentasikan perancangan Sub Bab 4.2.4 bagian 8.

The screenshot shows a web application window titled "Penentuan Besar Pinjaman". The window displays a form for entering loan data.

Selamat Datang, beta

**Form Data Peminjam**

Umur :  tahun

Pekerjaan :

Lama Pinjaman :  bulan

Gaji :  juta

Pinjaman yang diberikan  
 juta

**Gambar 5.17** Implementasi Antarmuka Halaman Pinjaman  
Sumber : implementasi