

**SISTEM PENUNJANG KEPUTUSAN KELAYAKAN PEMBIAYAAN  
USAHA KECIL MENENGAH (UKM) DENGAN MENGGUNAKAN  
LOGIKA FUZZY PADA BANK PEMBIAYAAN RAKYAT SYARIAH**

(Studi Kasus PT. BPRS WAKALUMI)



Disusun oleh :

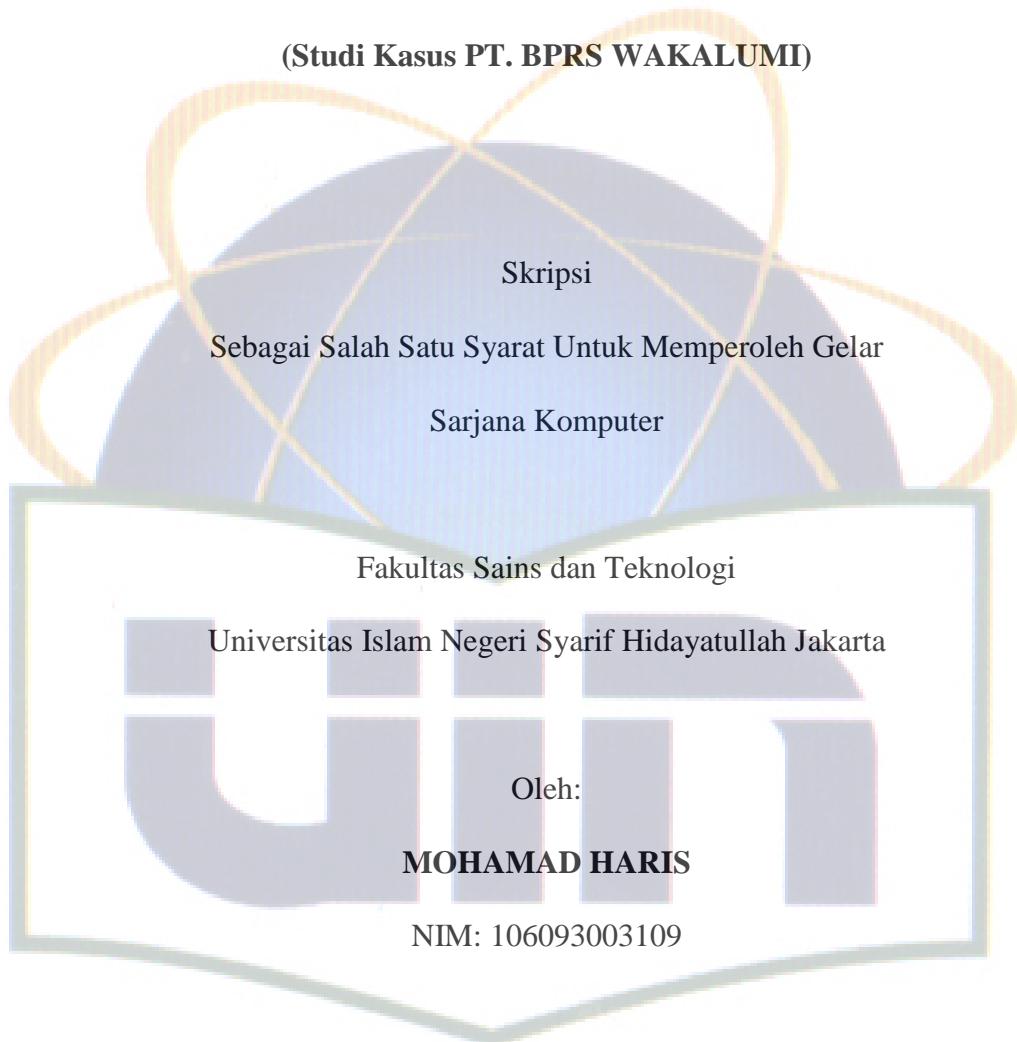
Mohamad Haris

NIM. 106093003109

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2011 M / 1432 H**

**SISTEM PENUNJANG KEPUTUSAN KELAYAKAN PEMBIAYAAN  
USAHA KECIL MENENGAH (UKM) DENGAN MENGGUNAKAN  
LOGIKA FUZZY PADA BANK PEMBIAYAAN RAKYAT SYARIAH**

**(Studi Kasus PT. BPRS WAKALUMI)**



**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH**  
**JAKARTA**  
**2011 M / 1432 H**

**SISTEM PENUNJANG KEPUTUSAN KELAYAKAN PEMBIAYAAN USAHA  
KECIL MENENGAH (UKM) DENGAN MENGGUNAKAN LOGIKA FUZZY  
PADA BANK PEMBIAYAAN RAKYAT SYARIAH**

(Studi Kasus: PT. BPRS WAKALUMI)

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Komputer  
Pada Fakultas Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh:

**MOHAMAD HARIS**

106093003109

Menyetujui,

Pembimbing I,

*Nur Aeni Hidayah*

Nur Aeni Hidayah, MMSI.  
NIP. 19750818 200501 2 008

Pembimbing II,

*Nia Kumaladewi*

Nia Kumaladewi, MMSI  
NIP. 19750412 200710 2 002

Mengetahui,

Ketua Program Studi Sistem Informasi

*Nur Aeni Hidayah*

Nur Aeni Hidayah, MMSI.  
NIP. 19750818 200501 2 008

## PENGESAHAN UJIAN

Skripsi yang berjudul “**SISTEM PENUNJANG KEPUTUSAN KELAYAKAN PEMBIAYAAN USAHA KECIL MENENGAH (UKM) DENGAN LOGIKA FUZZY PADA BANK PEMBIAYAAN RAKYAT SYARIAH**” telah diuji dan dinyatakan lulus dalam sidang munaqosyah Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada hari Selasa, 24 Mei 2011. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) Program Studi Sistem Informasi.

Jakarta, Juni 2011

Tim Pengaji,

Pengaji I

  
Zulfiandri, S.Kom, MMSI

NIP.19700130 200501 1 003

Pembimbing I

  
Nur Aeni Hidayah, MMSI

NIP.19750818 200501 2 008

Pengaji II

  
Ditdit Nugeraha Utama, MM, M.Com

NIP.19741129 200801 1 006

Pembimbing II

  
Nia Kumaladewi, MMSI

NIP.19754012 200710 2 002

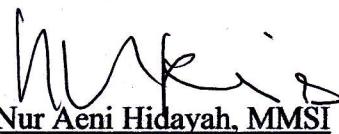
Mengetahui,



  
DR. Syopiansyah Jaya Putra, M.Sis

NIP.19680117 200112 1 001

Ketua Program Studi Sistem Informasi

  
Nur Aeni Hidayah, MMSI

NIP.19750818 200501 2 008

## **PERNYATAAN**

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSIINI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Jakarta, Juni 2011

**Mohamad Haris**

**106093003109**



## ABSTRAKSI

**MOHAMAD HARIS (106093003109)**, Sistem Penunjang Keputusan Kelayakan Pembiayaan Usaha Kecil Menengah (UKM) dengan Menggunakan Logika Fuzzy pada Bank Pembiayaan Rakyat Syariah (BPRS) - Studi Kasus PT.BPRS Wakalumi. (Dibawah bimbingan Ibu **NUR AENI HIDAYAH** dan **NIA KUMALADEWI** ).

Pemberian pembiayaan bagi pengusaha kecil tentu sangat membantu dalam perluasan usaha, hanya saja bagi pihak bank, namun kadangkala hal ini menimbulkan masalah seperti kasus pembiayaan bermasalah, penyebabnya tentu saja karena tunggakan dimana-mana. Untuk menghindari hal ini perbankan harus lebih teliti dalam menentukan pengusaha mana saja yang layak untuk memperoleh pembiayaan. Dalam Al-quran surat Al-Muzzamil ayat 29 “ ... dirikanlah sembahyang, tunaikanlah zakat dan berikanlah pinjaman kepada Allah pinjaman yang baik. ... ”. Selain itu, dalam melakukan proses penilaian kelayakan pembiayaan perbankan syariah masih menggunakan teknik skor, penggunaan teknik skoring akan menghasilkan nilai yang cenderung kaku (tidak fleksibel), sehingga menyebabkan para pengusaha sulit untuk mencapainya karena keterbatasan-keterbatasan yang dimilikinya. Oleh karena itu, diperlukan suatu *system* yang lebih fleksibel dalam melakukan penilaian. Teknik pengambilan keputusan pembiayaan dengan skoring yang lebih fleksible ini diharapkan dapat mampu membantu pihak BPRS dalam proses pengambilan keputusan pembiayaan UKM. *System* dapat diterapkan di BPRS (Bank Pembiayaan Rakyat Syariah). Aspek penilaian meliputi kinerja nasabah, usaha dan kepribadiannya. Data yang dibutuhkan *system* merupakan hasil penilaian petugas yang diperoleh dari data hasil *survey* di lapangan yang kemudian di input ke *system*. Peneliti mengembangkan *system* ini dengan terlebih dahulu menganalisa *system* yang sedang berjalan, kemudian mengembangkan *system* yang baru dengan menggunakan metode logika fuzzy. Peneliti menggunakan *Rapid Application System (RAD)* dengan *tools UML* sebagai pemodelannya. Untuk mendukung pengembangan *system* ini dengan menggunakan bahasa pemrograman *JAVA* dan *database MySQL*. Dengan diadakannya penelitian ini diharapkan pengembangan yang lebih dari *system* yang sudah dibuat, seperti penilaian kelayakan yang menyediakan berbagai kriteria yang sekiranya dibutuhkan oleh berbagai lembaga pemberi pembiayaan dan pinjaman.

**Kata Kunci :** SPK (Sistem Penunjang Keputusan), *Rapid Application Development (RAD)*, Fuzzy, Metode Tsukamoto, UML (*Unified Modelling Language*), *JAVA*, Pembiayaan.

V Bab + xxiii Hal + 159 Halaman + 32 Tabel + 54 Gambar + Pustaka + 8 Lampiran

Buku Acuan (24, 2001-2010)

## KATA PENGANTAR

*Bissmillahirrahmanirrahim*

Alhamdulillah, penulis panjatkan kepada Allah SWT atas seluruh rahmat dan karunia-Nya yang diberikan sehingga penulis dapat melaksanakan penelitian skripsi ini dan menyelesaikan penulisannya dengan lancar. Shalawat serta salam selalu tersampaikan kepada Rasulullah SAW yang telah menyampaikan risalah Islam sehingga dapat menyegarkan hati dalam menyelesaikan laporan ini.

Skripsi ini berjudul **“Sistem Penunjang Keputusan Kelayakan Pembiayaan Usaha Kecil Menengah (UKM) Dengan Menggunakan Logika Fuzzy Pada Bank Pembiayaan Rakyat Syariah (BPRS) - Studi Kasus PT.BPRS WAKALUMI.”**, yang disusun untuk memenuhi salah satu syarat dalam menyelesaikan program S1 pada Program Studi Sistem Informasi di Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah mendukung terselesaikannya skripsi ini. Karena tanpa dukungan dari mereka, penulis tidak akan mampu menyelesaikan laporan ini dengan baik. Mereka yang telah mendukung penulis adalah :

1. Bapak DR. Syopiansyah Jaya Putra, M.SIS, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Ibu Nur Aeni Hidayah, MMSI. selaku Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta sekaligus dosen pembimbing I yang telah membimbing penulis dalam menyelesaikan

skripsi ini dan Bapak Zainul Arham selaku sekretaris program studi Sistem Informasi.

3. Ibu Nia Kumaladewi, MMSI selaku pembimbing II penulis yang telah memberi banyak pengarahan bagi penulis dalam menyelesaikan skripsi ini.
4. Bapak Zulfiandry, MMSI Selaku Pengaji 1 yang telah memberi banyak masukan dalam revisi ba'da sidang.
5. Bapak Ditdit N. Utama, MMSI, M.Com Selaku Pengaji 2 yang telah memberi masukan dalam revisi.
6. Yang terkasih Ayahanda H.Rahmat dan Ibunda Hj.Umi Hani,  
*“Allahummaghfirlii waliwaalidayya warhamhummaa kamaa rabbayaanii shaghiiraa.”* Dan di lancarkan Rezeki.
7. Seluruh staf karyawan BPRS WAKALUMI, khususnya Kadiv Marketing yaitu Silvy dan Bagian Marketing Rasyidin, Rasyid, Mas Indra yang telah banyak membantu pengadaan data penelitian skripsi ini.
8. Yang terspesial saudara-saudariku yang sudah banyak membantu juga menemani hari-hariku, Semangatmu Ukhti Zahra dan khususnya untuk Subri dan Irwan (thank ilmu, informasi, tempat n Printernya), Egi, Ela, Amal, Dede, Agung, Fajar, yang telah membantu penulis baik secara langsung maupun tidak langsung. Jazakallah Khairan Katsiro aku cinta kalian karena Allah ^\_^.
9. Para Pendukung penulis dalam menyelesaikan pembuatan skripsi ini yang banyak direpotkan Azis, Intan, dan Bazel terima kasih banyak.

10. Teman-teman SI C angkatan 2006 yang tidak dapat disebutkan satu-persatu senangnya bisa mengenal dan dekat dengan kalian Seluruh penghuni kosan Irwan Tea, Kbm, iman, iqbali dan ali, tetap jalin silaturahim ya.
11. Sahabat-sahabatku di Team KKN, BEM FST 2008-2010, KOMDA FST, LDK Syahid dan PIM, Mujahid-Mujahidah Gd.1 yang selalu mendo'akan dan memberikan semangat baru ! *Jazakumullahu khairan katsira* atas ukhuwah terindah yang pernah ana rasakan.
12. Banyak lagi nama lain yang menghiasi halaman hati. Semoga doa *rabitah* senantiasa menjaga kita.

Layaknya tidak ada gading yang tak retak, begitu juga penulis yang tak luput dari kesalahan dalam penulisan ini. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan dari pembaca maupun pengguna skripsi ini. Kritik dan saran dapat disampaikan melalui [Mohamadhhs@gmail.com](mailto:Mohamadhhs@gmail.com) Semoga skripsi ini dapat berguna dan bermanfaat.

Jakarta, Juni 2011

Mohamad Haris  
106093003109

## DAFTAR ISI

Halaman Judul.....	i
Halaman Persetujuan Pembimbing.....	ii
Halaman Pengesahan.....	iii
Halaman Pernyataan.....	iv
Abstraksi.....	v
Kata Pengantar.....	vi
Daftar Isi.....	ix
Daftar Gambar.....	xv
Daftar Tabel.....	xviii
Daftar Simbol.....	xx
Daftar Istilah.....	xxii
Daftar Lampiran.....	xxiii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	6
1.6 Metode Penelitian.....	6
1.7 Sistematika Penulisan.....	8

## BAB III LANDASAN TEORI

2.1 Konsep Dasar Sistem.....	10
2.1.1 Definisi Sistem.....	10
2.1.2 Karakteristik Sistem.....	10
2.1.3 Klasifikasi Sistem.....	11
2.2 Konsep Dasar Informasi.....	11
2.2.1 Definisi Informasi.....	11
2.2.2 Kualitas Informasi.....	12
2.2.3 Nilai Informasi.....	13
2.3 Konsep Dasar Sistem Informasi.....	13
2.3.1 Pengertian Sistem Informasi.....	13
2.3.2 Komponen Sistem Informasi.....	14
2.4 Sistem Penunjang Keputusan.....	15
2.4.1 Tujuan Sistem Penunjang Keputusan.....	16
2.4.2 Struktur Masalah.....	17
2.4.3 Dukungan Keputusan.....	18
2.4.4 Karakteristik Dan Kemampuan SPK.....	18
2.4.5 Pengambilan Keputusan.....	19
2.4.6 Keuntungan Penggunaan Sistem Penunjang Keputusan.....	21
2.5 Pengertian Bank Pembiayaan Rakyat Syariah (BPRS).....	22
2.6 Konsep Pengambilan Keputusan Pembiayaan UKM.....	23
2.6.1 Pengertian Usaha Kecil Menengah (UKM).....	23
2.6.2 Pengertian Pembiayaan.....	25
2.6.2.1 Jenis-Jenis Pembiayaan.....	27

2.6.2.2 Pembiayaan Menurut Jangka Waktu.....	27
2.6.2.3 Pembiayaan Menurut Segmentasi Pasar.....	27
2.6.3 Tujuan Dan Fungsi Pembiayaan.....	27
2.6.4 Prosedur Dalam Pemberian Pembiayaan.....	28
2.6.5 Prinsip-Prinsip Pemberian Pembiayaan.....	30
<i>2.7 Artificial Intelligence.....</i>	33
2.8 Logika Fuzzy.....	34
2.8.1 Fungsi Keanggotaan .....	38
2.8.2 Representasi Kurva Segitiga.....	40
2.8.3 Representasi Kurva Bentuk Bahu.....	41
<i>2.9 Fuzzy Inference System.....</i>	41
2.9.1 Metode Tsukamoto.....	42
2.9.2 Metode Mamdani.....	42
2.9.3 Metode Sugeno.....	42
2.10 Tools Perancangan Sistem <i>Unified Modelling Language (UML)</i> ....	45
2.9.4 Konsep Untuk Pemodelan Objek.....	45
2.9.5 Tipe Diagram <i>UML</i> .....	46
2.11 Bahasa Pemrograman .....	48
2.11.1 Java.....	48
2.11.2 Sejarah Java.....	48
2.11.3 Karakteristik Java.....	50
2.11.4 Fitur-Fitur Java.....	51
2.12 MySQL.....	52
2.12.1 Tipe Data Dalam MySQL.....	53

2.13Database.....	54
2.13.1 Komponen <i>Database</i> .....	55
2.13.2 Kegunaan <i>Database</i> .....	56
2.14Jaringan Komputer	
2.14.1 Definisi Jaringan Komputer.....	56
2.14.2 Arsitektur Jaringan Komputer.....	56
2.14.3 Definisi <i>Intranet</i> .....	58
2.15EvaluasiKelayakanPembangunanYangLain.....	58
2.16Penelitian Sejenis.....	60
 <b>BAB III METODOLOGI PENELITIAN</b>	
3.1 Pengumpulan Data.....	66
3.1.1 Observasi.....	66
3.1.2 Interview/Wawancara.....	71
3.1.3 Studi Pustaka.....	71
3.2 Pengembangan Sistem.....	72
3.2.1 Fase Perencanaan Syarat-Syarat.....	74
3.2.2 Fase Workshop <i>Design</i> .....	74
3.2.3 Fase Implementasi.....	75
3.3 Kerangka Berfikir.....	75

## **BAB IV ANALISA DAN PEMBAHASAN**

4.1 Perencanaan Syarat-Syarat.....	77
4.1.1 Identifikasi Masalah.....	77

4.1.2 Analisis Sistem Yang Berjalan.....	82
4.1.3 Tujuan Pengembangan <i>Prototype System</i> .....	83
4.1.4 Kebutuhan <i>Prototype</i> Sistem.....	84
4.2 Fase <i>Workshop Design</i> .....	85
4.2.1 Sistem Yang Di Usulkan.....	85
4.2.2 Perancangan Proses Pada SPK Kelayakan UKM.....	87
4.2.2.1 Karakter Model Secara Fungsional dan Operasional....	87
4.2.2.2 Melakukan DekomposisiMenjadi Himpunan <i>Fuzzy</i> .....	90
4.2.2.3 Membuat Aturan <i>Fuzzy</i> .....	93
4.2.2.4 Menentukan Metode <i>Defuzzy</i> Untuk Setiap Variabel...	95
4.2.2.5 Menjalankan Simulasi Sistem dan Pengujian.....	98
4.2.3 Rancangan Spesifikasi SPK.....	109
4.2.3.1 Identifikasi <i>Actor</i> .....	109
4.2.3.2 <i>Usecase Diagram</i> .....	109
4.2.3.3 <i>Activity Diagram</i> .....	115
4.2.3.4 <i>Statechart Diagram</i> .....	122
4.2.3.5 <i>Sequence Diagram</i> .....	124
4.2.3.6 <i>Class Diagram</i> .....	130
4.2.4 Rancangan <i>Database</i> .....	131
4.2.4.1 Nama <i>Database</i> .....	131
4.2.4.2 Identifikasi <i>Database</i> .....	131
4.2.4.3 <i>Physical Database Schema</i> .....	135
4.2.5 Rancangan Antar Muka Pemakai.....	139
4.3 Fase Implementasi.....	146

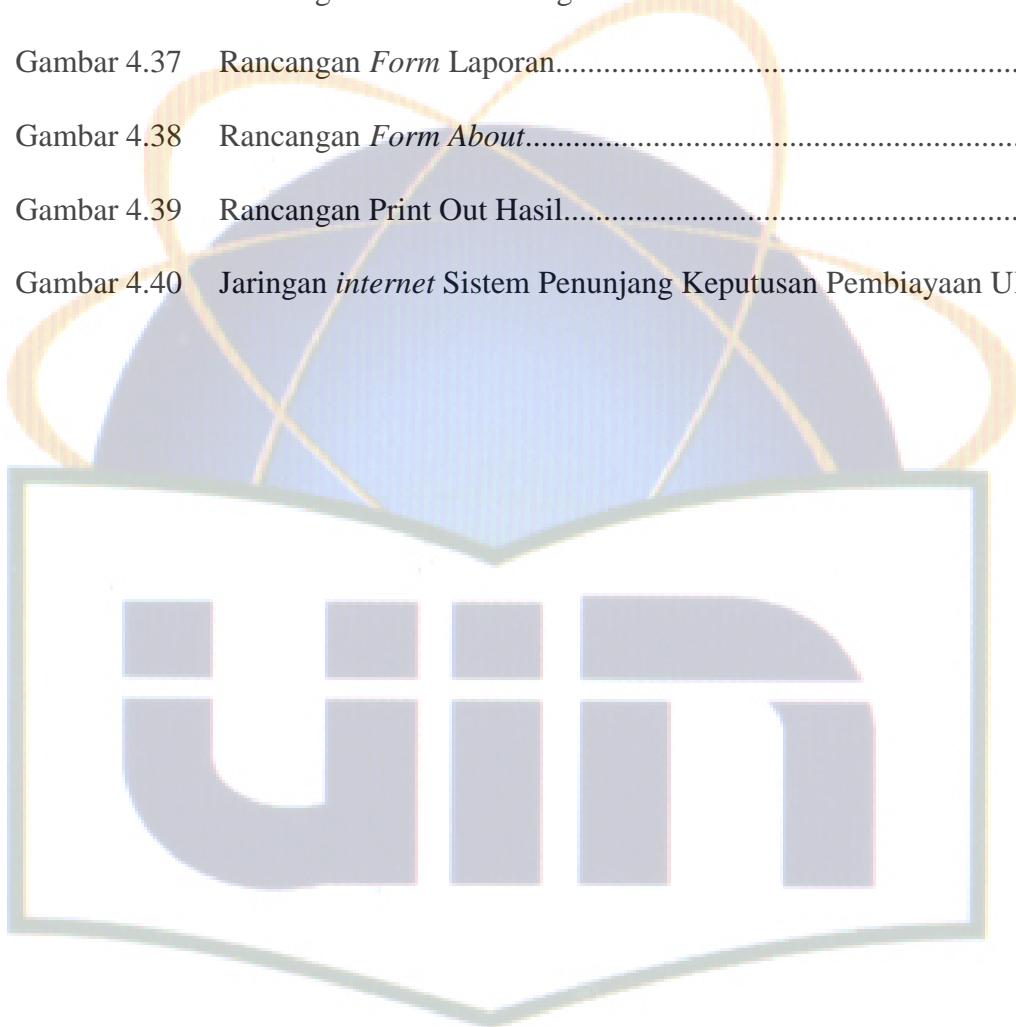
4.3.1 Konstruksi Sistem Penunjang Keputusan.....	146
4.3.2 Pengujian Sistem Penunjang Keputusan.....	146
4.3.3 Sarana Pendukung Sistem.....	150
4.3.4 Kelebihan.....	152
4.3.5 Kekurangan.....	152
<b>BAB V KESIMPULAN DAN SARAN</b>	
5.1 Kesimpulan.....	154
5.2 Saran.....	155
<b>DAFTAR PUSTAKA.....</b>	156
Lampiran.....	162

## DAFTAR GAMBAR

Gambar 2.1	SPK Berfokus pada Masalah-masalah semi terstruktur.....	17
Gambar 2.2	Karakteristik dan Manfaat SPK.....	19
Gambar 2.3	Contoh Pemetaan <i>Input-Output</i> .....	35
Gambar 2.4	Representasi Linear Naik.....	39
Gambar 2.5	Representasi Linear Turun.....	40
Gambar 2.6	Representasi Kurva Segitiga.....	40
Gambar 2.7	Representasi Kurva Bahu.....	41
Gambar 2.8	Inferensi dengan menggunakan Metode Tsukamoto	43
Gambar 3.1	Struktur Organisasi BPRS Wakalumi.....	71
Gambar 3.2	Fase-fase <i>RAD</i> .....	74
Gambar 3.3	Kerangka Berfikir.....	76
Gambar 4.1	<i>Rich Picture</i> Analisa Sistem yang Berjalan.....	83
Gambar 4.2	<i>Rich Picture</i> Sistem yang Diusulkan.....	86
Gambar 4.3	Fungsi Keanggotaan Variabel <i>Input</i> .....	91
Gambar 4.4	Fungsi Keanggotaan Variabel <i>Output</i> .....	93
Gambar 4.5	Fungsi Keanggotaan Variabel Jaminan pada contoh Tabel 4.6.....	99
Gambar 4.6	Fungsi Keanggotaan Variabel Usaha pada contoh Tabel 4.6.....	100
Gambar 4.7	Fungsi Keanggotaan Variabel Kemampuan pada Contoh Tabel 4.6...	101
Gambar 4.8	Fungsi Keanggotaan Variabel Karakter pada Contoh Tabel 4.6.....	102
Gambar 4.9	Fungsi Keanggotaan Variabel Evaluasi Penilaian Contoh Tabel 4.6..	103
Gambar 4.10	<i>Use Case Diagram</i> Sistem Penunjang Keputusan yang Diajukan ....	109

Gambar 4.11	<i>Activity Diagram</i> dari <i>Use Case Login</i> .....	115
Gambar 4.12	<i>Activity Diagram</i> dari <i>Use Case Mengisi Kriteria</i> .....	116
Gambar 4.13	<i>Activity Diagram</i> dari <i>Use Case Mengisi Nilai Kriteria</i> .....	117
Gambar 4.14	<i>Activity Diagram</i> dari <i>Use Case Pemeriksaan</i> .....	118
Gambar 4.15	<i>Activity Diagram</i> dari <i>Use Case Perhitungan</i> .....	119
Gambar 4.16	<i>Activity Diagram</i> dari <i>Use Case Mengisi Data Calon Debitur</i> .....	120
Gambar 4.17	<i>Statechart Diagram</i> dari <i>Use Case Login</i> .....	122
Gambar 4.18	<i>Statechart Diagram</i> dari <i>Use Case Mengisi Kriteria</i> .....	122
Gambar 4.19	<i>Statechart Diagram</i> dari <i>Use Case Informasi Calon Debitur</i> .....	123
Gambar 4.20	<i>Sequence Diagram</i> Dari <i>Use Case Login</i> .....	124
Gambar 4.21	<i>Sequence Diagram</i> dari <i>Use Case Mengisi Kriteria</i> .....	125
Gambar 4.22	<i>Sequence Diagram</i> dari <i>Use Case Mengisi Nilai Kriteria</i> .....	126
Gambar 4.23	<i>Sequence Diagram</i> dari <i>Use Case Pemeriksaan</i> .....	127
Gambar 4.24	<i>Sequence Diagram</i> dari <i>Use Case Perhitungan</i> .....	128
Gambar 4.25	<i>Sequence Diagram</i> dari <i>Use Case Mengisi Data Calon Debitur</i> .....	129
Gambar 4.26	<i>Class Diagram</i> .....	130
Gambar 4.27	<i>Physical Database Schema</i> .....	135
Gambar 4.28	Rancangan <i>Form</i> Masuk.....	138
Gambar 4.29	Rancangan <i>Form</i> Login.....	138
Gambar 4.30	Rancangan <i>Form</i> Utama Admin.....	139
Gambar 4.31	Rancangan <i>Form</i> Pemohon.....	139
Gambar 4.32	Rancangan <i>Form</i> Utama Manager.....	140
Gambar 4.33	Rancangan <i>Form</i> Nilai Kriteria.....	141

Gambar 4.34	Rancangan <i>Form</i> Kriteria.....	141
Gambar 4.35	Rancangan <i>Form</i> Pemeriksaan.....	142
Gambar 4.36	Rancangan <i>Form</i> Perhitungan.....	143
Gambar 4.37	Rancangan <i>Form</i> Laporan.....	144
Gambar 4.38	Rancangan <i>Form</i> <i>About</i> .....	144
Gambar 4.39	Rancangan Print Out Hasil.....	145
Gambar 4.40	Jaringan <i>internet</i> Sistem Penunjang Keputusan Pembiayaan UKM...	152



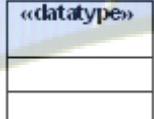
## DAFTAR TABEL

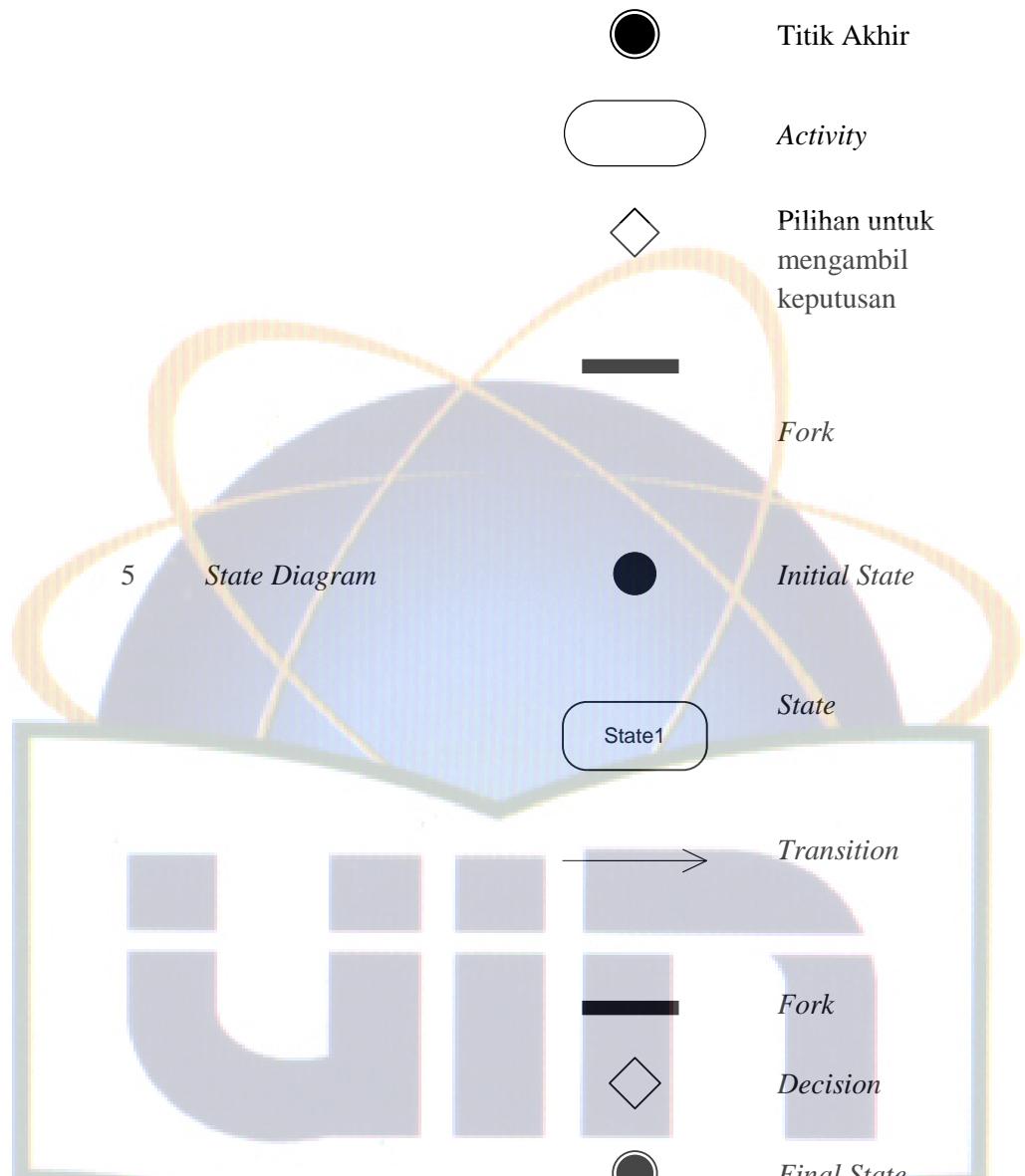
Tabel 2.1 Perbandingan Nilai Uang dalam UKM .....	24
Tabel 2.2 Produksi Makanan Kaleng Jenis ABC.....	44
Table 2.3 <i>Output</i> Jumlah Produksi dengan 3 Metode <i>Fuzzy</i> .....	44
Tabel 2.4 Hasil Penelitian Sejenis.....	62
Tabel 2.5 Hasil Penelitian Sejenis.....	63
Tabel 2.6 Hasil Penelitian Sejenis.....	65
Tabel 4.1 Kriteria Syarat Jaminan.....	80
Tabel 4.2 Variabel Penilaian Kelayakan Pembiayaan Beserta <i>Range</i> .....	88
Tabel 4.3 Variabel <i>Input</i> Beserta <i>Range</i> .....	91
Tabel 4.4 Klasifikasi Nilai Hasil Evaluasi Penilaian .....	93
Tabel 4.5 Aturan <i>Fuzzy</i> Berdasarkan Proses Bisnis Di BPRS WAKALUMI.....	95
Tabel 4.6 Kombinasi Masukan Calon Debitur 1 – Pedagang Daging.....	98
Tabel 4.7 Simulasi dengan Berbagai Variasi Masukan dan Keluaran.....	108
Tabel 4.8 Identifikasi <i>Actor</i> .....	109
Tabel 4.9 Identifikasi <i>Use Case Diagram</i> .....	110
Tabel 4.10 Narasi <i>Use Case Login</i> .....	110
Tabel 4.11 Narasi <i>Use Case Mengisi Kriteria</i> .....	111
Tabel 4.12 Narasi <i>Use Case Mengisi Nilai Kriteria</i> .....	112
Tabel 4.13 Narasi <i>Use Case Melakukan Pemeriksaan</i> .....	112
Tabel 4.14 Narasi <i>Use Case Melakukan Perhitungan</i> .....	113
Tabel 4.15 Narasi <i>Use Case Mengisi Informasi Calon Debitur</i> .....	114

Tabel 4.16 Tabel Identifikasi <i>Database</i> .....	131
Tabel 4.17 Tabel Login.....	132
Tabel 4.18 Tabel Kriteria Nilai.....	132
Tabel 4.19 Tabel Kriteria Variabel.....	133
Tabel 4.20 Tabel Master Perhitungan.....	133
Tabel 4.21 Tabel Pemohon.....	133
Tabel 4.22 Tabel Hasil Perhitungan.....	134
Tabel 4.23 Tabel <i>CRUD Database</i> .....	136
Tabel 4.24 Perbandingan dalam Pengujian Sistem Penilaian Teknik Skoring dan Logika Fuzzy.....	146
Tabel 4.25 Pengujian Dengan Blackbox Testing.....	148

## DAFTAR SIMBOL

Diagram UML (*Unified Modelling Language*)

No	Nama Diagram	Simbol	Nama
1	Use Case Diagram		Actor
			Use Case
			Participant
2	Class Diagram		Datatype
			Interface
			Generalization
			Participant
3	Sequence Diagram		Simpel Message
			Synchronous
			Asynchronous
4	Activity Diagram		Titik Awal



Sumber: Munawar, 2005

## DAFTAR ISTILAH

No	Istilah	Arti
1	Ijarah Muntahiya Bit Tamlik	Akad sewa beli
2	Mudharabah	pembiayaan untuk modal kerja usaha, bagi hasil sesuai kesepakatan bersama
3	Musyarakah	kerjasama yang dilakukan dua orang atau lebih untuk mengikat diri dalam perserikatan modal dan keuntungan
4	Al-Murabahah	suatu perjanjian pembiayaan berdasarkan sistem jual beli, dimana bank membiayai (membelikan) kebutuhan investasi nasabah dan menjual kembali kepada nasabah ditambah dengan keuntungan yang disepakati.
5	Al Qardul Hasan	Akad pinjaman murni yang diberikan kepada peminjam (muqtaridh) selama waktu tertentu dan dikembalikan dalam jumlah yang sama pada saat jatuh tempo
6	Ijarah	Sewa
7	Akad	Ijab Qabul
8	Mal	Modal, harta
9	Dharabah	Kerja atau usaha
10	Qardh	Pinjaman
11	Character	Karakter
12	Capacity	Kemampuan
13	Capital	Permodalan
14	Colleteral	Jaminan
15	Condition	kondisi ekonomi
16	Personality	Kepribadian
17	Part	Bagian
18	Purpose	Tujuan
19	<i>Prospect</i>	Prospek
20	Payment	Pembayaran
21	<i>Profitability</i>	Keuntungan
22	Protection	Perlindungan
23	Input	Masuk
24	Output	Keluar

## DAFTAR LAMPIRAN

Lampiran A Tampilan Aplikasi.....	A
Lampiran B Wawancara.....	B
Lampiran C Surat Keterangan.....	C
Lampiran D Formulir Pendaftaran Pembiayaan .....	D
Lampiran E Tabel Aturan ( <i>Rule</i> ) Penilaian Kelayakan Pembiayaan.....	E
Lampiran F Kriteria Dan Bobot Penilaian Kelayakan Calon Debitur.....	F
Lampiran G Tabel Variabel <i>Input</i> dan <i>Output</i> .....	G
Lampiran H <i>Source Code</i> .....	H



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Dalam perekonomian Indonesia, sektor Usaha Kecil Menengah biasa disebut UKM memegang peranan yang sangat penting terutama apabila dikaitkan dengan jumlah tenaga kerja yang mampu diserap oleh UKM karena tenaga kerjanya mencapai 91.8% atau 97.3% dan UKM ini selain memiliki arti strategis bagi pembangun, juga sebagai upaya untuk memeratakan hasil pembangunan (Asnur. 2009). Penelitian mengenai Usaha Kecil Menengah (UKM) telah banyak dilakukan oleh berbagai pihak diantaranya: Sektor penting dalam perekonomian Indonesia, UKM mendominasi kegiatan usaha misalnya disektor pertanian lebih dari 99 persen kegiatan usaha dilakukan oleh pengusaha kecil (Anoraga, *et, al.* 2002). Namun sayangnya UKM masih memiliki kelemahan. Oleh karena itu, perhatian yang besar perlu dicurahkan dalam hal pemberdayaan usaha ini. Studi peningkatan peran Bank perkreditan rakyat (BPR) dalam usaha mikro kecil (UMK) di Sumatra Barat (Herri. 2008). Selain itu ada juga penelitian yang mengkaji tindakan peningkatan peran koperasi dan UKM sebagai lembaga keuangan alternatif (Situmorang. 2005), pengkajian Usaha Mikro Indonesia (Joko. 2006). Pengaruh Faktor-Faktor Eksternal dan Internal Terhadap Kinerja Usaha Mikro dan Kecil (UMK) di Sulawesi Selatan (Musran. 2010), Sistem Informasi Pembiayaan *Mudharabah* untuk Pemberian Modal bagi Usaha Kecil Menengah (UKM) (Sari. 2011).

Seiring perkembangan ilmu pengetahuan dan Teknologi Penelitian mengenai sistem pendukung keputusan telah banyak dilakukan oleh berbagai pihak diantaranya pada dalam dunia perbankan. Dalam perbankan ada penelitian penentu kelayakan pada UKM dengan judul Penyusunan Decision Support System (DSS) Studi Kelayakan Ekonomi dan Finansial bagi UMKM (Asnur. 2009). Adanya penelitian tentang Fuzzy dengan judul Sistem Pendukung Keputusan Untuk Menentukan Penerima Beasiswa Bank BRI Menggunakan FMADM (Wibowo *et, al.* 2009). Dalam dunia pendidikan Khoiruddin (Khoiruddin. 2008) melakukan penelitian dengan judul Sistem Pendukung Keputusan Kelayakan Calon Rintisan Sekolah Bertaraf Internasional dengan Metode Fuzzy Associative Memory. Ada pula penelitian pada handphone dengan judul Model Fuzzy Tahani untuk Pemodelan Sistem Pendukung Keputusan (SPK) Kasus: Rekomendasi Pembelian Handphone (Amalia. 2010). Juga melakukan penelitian dengan Fuzzy yang berjudul Sistem Inferensi Fuzzy Untuk Menentukan Tingkat Resiko Penyakit Geriatri Melalui Aplikasi Berbasis Web (Rahmayani. 2008). Usaha Peningkatan Kualitas Pelayanan Dengan Pendekatan Fuzzy Dan Metode Service Quality Pada Pusat Perbelanjaan Assalam Hypermarket (Nurdiyanto. 2008). Model Fuzzy Dengan Metode Fuzzy (Widhiastiwi. 2007). Aplikasi Logika Fuzzy Metode Tsukamoto Untuk Memprediksi Potensi Serangan Stroke. (Emi *et, al.* 2009). Penentuan Tingkat Resiko Penyakit Menggunakan Tsukamoto Fuzzy Inference System oleh (Kusumadewi. 2009).

Ada pula dengan system Skoring seperti penelitian yang berjudul *An Introduction to Credit Scoring For Small and Medium Size Enterprises* (Marquez. 2008). Penelitian dengan Skoring juga digunakan yang berjudul *Pedoman Monitoring Dan Evaluasi Dosen Inti Penelitian Universitas Indonesia* (DRPM UI, 2009).

Dalam Al-Qur'an Surat Al-Muzammil ayat 20 "... dirikanlah sembahyang, tunaikanlah zakat dan berikanlah pinjaman kepada Allah pinjaman yang baik ....". Maka Salah satu bentuk perhatian yang diperlukan oleh UKM adalah mudharabah (pembiayaan modal). Pemberian modal diwujudkan dengan cara memberikan pinjaman oleh perbankan syariah.

Pemberian pembiayaan bagi pengusaha tentu sangat membantu dalam perluasan usaha. Hanya saja bagi pihak bank, hal ini dapat menimbulkan masalah yang tidak mudah untuk dipecahkan. Terbukti dengan banyaknya pembiayaan bermasalah. Penyebabnya karena pembiayaan yang telah di keluarkan tidak dibayarkan dalam jangka waktu tertentu. Sehingga terjadi penunggakan dimana-mana.

Untuk menghindari hal ini perbankan harus lebih teliti dalam menentukan pengusaha mana saja yang layak untuk memperoleh pembiayaan. Hal ini ditujukan agar memperoleh penilaian tepat dan akurat.

Selain itu, dalam melakukan proses pengambilan keputusan perbankan masih menggunakan teknik skoring, penggunaan teknik skoring akan menghasilkan nilai tidak fleksibel, sehingga menyebabkan para pengusaha

sulit untuk mencapainya karena keterbatasan yang dimilikinya. Nilai dari perhitungan skoring itu hanya memiliki satu nilai kemungkinan tapi dengan menggunakan metode *Fuzzy*, ada beberapa kemungkinan nilai yang akan dihasilkan, sehingga lebih menguntungkan calon debitur. Oleh karena itu, diperlukan suatu sistem yang lebih fleksibel dalam melakukan penilaian.

Metode *Fuzzy* merupakan bagian dari salah satu bidang ilmu komputer yaitu *artificial intelligent* (kecerdasan buatan). Metode *Fuzzy* diformulasikan dalam rangka mencari nilai tengah antara bilangan 0 dan 1. Hal ini seiring dengan usaha untuk membuat komputer yang bekerja seperti cara manusia berfikir, sebab komputer pada dasarnya adalah sebuah mesin hitung yang tidak berfikir (Kusumadewi, 2004). Agar permasalahan tersebut dapat diatasi, maka penulis mencoba membangun “Sistem Penunjang Keputusan Pembiayaan Kelayakan Usaha Kecil Menengah dengan Menggunakan Logika *Fuzzy* pada Bank Pembiayaan Rakyat Syariah” dengan adanya penelitian ini diharapkan mampu membantu peran manajer dalam proses pengambilan keputusan UKM.

Sistem ini dapat diterapkan di BPRS (Bank Pembiayaan Rakyat Syariah). Dikarenakan penilaian ini meliputi kinerja nasabah dan usaha. Data yang dibutuhkan dalam sistem penunjang keputusan ini merupakan hasil penilaian petugas yang diperoleh dari hasil *survey* di lapangan yang kemudian di *input* ke dalam Sistem.

## 1.2 Rumusan Masalah

Dari beberapa permasalahan yang ada, maka dapat dirumuskan,

1. Bagaimana penerapan Logika *Fuzzy* dalam pengembangan Sistem Penunjang Keputusan Kelayakan Pembiayaan pada UKM?
2. Bagaimana membantu *manager* mengambil keputusan kelayakan calon debitur yang lebih akurat.

## 1.3 Batasan Masalah

Agar pembahasan dalam skripsi ini tidak meluas, namun dapat mencapai hasil yang lebih optimal, dalam penulisan skripsi ini penulis membatasi masalah pada:

1. Sistem penunjang keputusan kelayakan pembiayaan yang berlaku di BPRS WAKALUMI untuk UKM berdasarkan kriteria yaitu : Jaminan, Usaha, Kemampuan dan Karakter.
2. Teknik Penilaian kelayakan pembiayaan dengan logika *Fuzzy* menggunakan metode Tsukamoto.

## 1.4 Tujuan Penelitian

1. Membantu peran *manager* dalam melakukan pengambilan keputusan penilaian kelayakan pembiayaan UKM berdasarkan kriteria yang berlaku BPRS WAKALUMI.
2. Membuat Sistem Penunjang Keputusan Kelayakan Pembiayaan UKM pada Bank Pembiayaan Rakyat Syariah.

## 1.5 Manfaat Penelitian

1. Memberikan gambaran umum mengenai pembiayaan yang ada di lembaga keuangan syariah khususnya tentang pengambilan keputusan.
2. Dapat memberikan pemahaman mengenai logika *fuzzy* dapat digunakan pada suatu kasus penunjang keputusan.
3. Memperkaya khazanah keilmuan di bidang Sistem Informasi khususnya dibidang Bisnis Syariah.
4. Dapat dijadikan referensi untuk bahan perkuliahan khususnya pada bidang peminatan Sistem Informasi Bisnis Syariah.

## 1.6 Metode Penelitian

Pada penelitian skripsi ini, diperlukan data-data yang lengkap sebagai bahan yang dapat mendukung kebenaran materi uraian dan pembahasan oleh karena itu diperlukan metodologi pengumpulan data dan metode pengembangan sistem.

### 1.6.1 Metode Pengumpulan Data

Dalam penyusunan skripsi ini, diperlukan data informasi yang lengkap untuk keperluan pengembangan sistem. Oleh karena itu sebelum menyusun skripsi, dilakukan persiapan terlebih dahulu yaitu riset atau penelitian. Hal ini ditujukan untuk menjaring data-data atau bahan materi yang diperlukan.

Adapun metode pengumpulan data-data yang dilakukan adalah sebagai berikut:

## 1. Metode Observasi

Observasi adalah metode pengumpulan informasi dengan cara pengamatan atau peninjauan langsung terhadap sistem yang berjalan.

## 2. Metode *Interview*

Pengumpulan data dan informasi dengan cara melakukan wawancara secara langsung dengan manajer dan serta pihak BPRS lainnya yang terkait untuk mengetahui gambaran proses penilaian pengambilan keputusan pembiayaan.

## 3. Metode Pustaka

Pengumpulan data dan informasi dengan cara menelaah referensi yang dijadikan acuan pembahasan dalam masalah ini.

### 1.6.2 Metode Pengembangan Sistem

Yaitu dengan mengembangkan metode *prototype* sistem dengan menggunakan *Rapid Application Developmen (RAD)*, dengan menggunakan *tools* pengembangan Sistem *Unified Modelling Model (UML)* sebagai modelnya. *RAD* yang memiliki tahapan berikut (Kendall dan Kendall, 2003):

#### 1. Perancangan Syarat-Syarat

Fase ini pengguna dan manajer bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta mengidentifikasi syarat-syarat yang ditimbulkan dari tujuan tersebut. Fase ini memerlukan peran aktif dari kedua kelompok tersebut. Selain itu juga melibatkan penggunaan dari beberapa level yang berbeda dalam organisasi.

## 2. Workshop Design

Dalam fase ini pengguna dan manajer bersama-sama merancang dan memperbaiki apa yang kemudian dapat digambarkan selama *Workshop* dalam selama *workshop design RAD*. Pengguna merespon *working prototype* yang ada dan manajer memperbaiki modul-modul yang dirancang menggunakan perangkat lunak berdasarkan respon pengguna dan pemodelannya dengan *UML*.

## 3. Fase Implementasi

Manajer bekerja secara *intens* dengan pengguna selama *Workshop design* untuk merancang aspek-aspek bisnis dan *non teknis* dari perusahaan. Segera setelah aspek-apek ini disetujui sistem dibangun dan disaring, sub-sub sistem diuji coba dan diperkenalkan kepada organisasi.

### 1.7 Sistematika Penulisan

Dalam menyelesaikan skripsi ini penulisan menyajikan tulisan ini menjadi beberapa bab yaitu:

#### BAB I PENDAHULUAN

Bab ini membahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, metode penelitian dan sistematika penulisan.

#### BAB II LANDASAN TEORI

Dalam bab ini berisi uraian tentang landasan teori yang diperlukan dalam sistem penunjang pengambilan keputusan pembiayaan pada

usaha kecil menengah. Teori-teori tersebut antara lain konsep Bank Pembiayaan Rakyat Syariah (BPRS), konsep penilaian pengambilan keputusan pembiayaan, sistem skoring, *database*, serta teori pendukung lainnya.

### BAB III METODE PENELITIAN

Bab ini menguraikan secara rinci metodologi yang digunakan dalam manajer dan design yang akan dibuat, metodologi tersebut meliputi metode pengumpulan data dengan cara observasi langsung ke BPRS WAKALUMI, metode wawancara dengan pihak BPRS dan studi pustaka yang berkaitan dengan pengembangan *prototype* penilaian pengambilan keputusan pembiayaan UKM. Selain itu pengembangan *prototype* sistem digunakan *Rapid Application Development (RAD)* dengan *tools UML (Unified Modelling Model)*.

### BAB IV ANALISA DAN PEMBAHASAN

Bab ini membahas lebih rinci implementasi metodologi *RAD* dalam mengembangkan *prototype* sistem penunjang keputusan penilaian kelayakan pembiayaan UKM secara syariah pada BPRS WAKALUMI.

### BAB V PENUTUP

Bab ini berisi kesimpulan dari seluruh bab dan saran untuk pengembangan sistem lebih lanjut.

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Dasar Sistem

##### 2.1.1 Definisi Sistem

Sistem adalah kumpulan dari komponen atau elemen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu (Jogiyanto, 2005).

Sistem adalah suatu kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu (Wahyono, 2003).

Pengertian sistem dilihat dari sudut pandang sistem informasi yang berorientasi *object* adalah sekumpulan komponen yang mengimplementasikan model dan fungsionalitas yang dibutuhkan, komponen tersebut saling berinteraksi didalam sistem guna mentransformasi *input* yang diberikan kepada sistem tersebut menjadi *output* yang berguna dan bernilai bagi *actor*-nya (Irwanto, 2006).

##### 2.1.2 Karakteristik Sistem

Berikut adalah karakteristik sistem yang dapat dapat dibedakan satu sistem dengan sistem lainnya (Al Fatta, 2007):

- a. Batasan (*boundary*)
- b. Lingkungan (*environment*)

- c. Masukan (*input*)
- d. Keluaran (*output*)
- e. Komponen (*component*)
- f. Penghubung (*interface*)
- g. Penyimpanan (*storage*)

### 2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari berbagai sudut pandang, diantaranya sebagai berikut (Jogiyanto, 2005):

- a. Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*)
- b. Sistem alamiah (*natural system*) dan sistem buatan (*human made system*)
- c. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*)
- d. Sistem tertutup (*clossed system*) dan sistem terbuka (*open system*)

## 2.2 Konsep Dasar Informasi

### 2.2.1 Definisi Informasi

Informasi dapat didefinisikan sebagai hasil dari pengelolaan data dalam bentuk lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian nyata yang digunakan untuk pengambilan keputusan (Jogiyanto, 2005).

Informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian–kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan (Wahyono, 2003).

## 2.2.2 Kualitas Informasi

Menurut (Wahyono, 2003) Kualitas informasi sangat dipengaruhi oleh 3 hal, yaitu:

a. Relevansi

Informasi dinyatakan berkualitas jika relevan bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya.

b. Akurasi

Sebuah informasi dapat dikatakan akurat jika informasi tersebut tidak menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya.

c. Tepat Waktu

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat. Informasi yang terlambat tidak akan mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar pengambilan keputusan dapat menimbulkan kesalahan dalam tindakan yang akan diambil.

### 2.2.3 Nilai Informasi

Nilai dari informasi ditentukan oleh dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu sistem dikatakan bernilai bila manfaatnya lebih efektif dibandingkan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak di dalam perusahaan (Jogiyanto, 2005).

## 2.3 Konsep Dasar Sistem Informasi

### 2.3.1 Pengertian Sistem Informasi

Sistem informasi adalah suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan (Wahyono, 2003).

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

### 2.3.2 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai sebuah sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya (Jogiyanto, 2005).

#### 1. Blok Masukan (*Input*)

*Input* mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

#### 2. Blok Model

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

#### 3. Blok Keluaran (*Output*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

#### 4. Blok Teknologi

Teknologi merupakan “kotak alat” (*toolbox*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model,

menyimpan data mengakses data, menghasilkan dan mengirimkan keluaran dan bantuan pengendalian dari sistem secara keseluruhan.

#### 5. Blok Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

#### 6. Blok Kendali

Untuk upaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya.

### 2.4 Sistem Penunjang Keputusan

Menurut Keen dan Morton, sistem penunjang keputusan adalah perpaduan antara sumber daya intelektual manusia dan kemampuan komputer yang berbasis komputer untuk meningkatkan kualitas keputusan dalam mendukung pengambilan keputusan yang berhubungan dengan masalah semi terstruktur. (Turban, 2005)

Turban dalam bukunya mengemukakan bahwa “*A DSS is an interactive, flexible, and adaptable CBIS, specially develop for supporting the solution of a particular management problem for improved decision making*”.

Sistem penunjang Keputusan adalah sistem informasi berbasis komputer interaktif, fleksibel, dan disesuaikan yang dikembangkan secara

khusus untuk mendukung dalam memecahkan masalah, dari masalah manajemen tertentu untuk meningkatkan pengambilan keputusan yang lebih baik. (Turban,2005)

Dengan demikian Sistem Penunjang Keputusan merupakan sistem yang berbasis komputer yang mudah dalam penggunaannya, lebih memfokuskan dalam pengambilan keputusan agar lebih efektif dan dapat memecahkan masalah semi terstruktur.

#### 2.4.1 Tujuan Sistem Penunjang Keputusan

Menurut Peter G.W Keen dan Scott Morton ada tiga tujuan yang harus dicapai oleh sistem penunjang keputusan, yaitu:

1. Membantu manajer membuat keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung penilaian manajer bukan mencoba untuk mengantikannya.
3. Meningkat efektifitas pengambilan keputusan manajer dari pada efisiensinya.

Tujuan-tujuan ini berhubungan dengan tipe prinsip dasar dari konsep SPK ialah struktur masalah, dukungan keputusan, dan efektivitas keputusan (McLeod, 2004)

### 2.4.2 Struktur Masalah

Struktur masalah dapat mempengaruhi pemecahan masalah, *Decision Support System* diarahkan untuk mendukung terhadap pemecahan masalah semi terstruktur. Untuk melihat secara jelas mengenai masalah ada tiga pengelompokan masalah yaitu:

1. Masalah terstruktur.

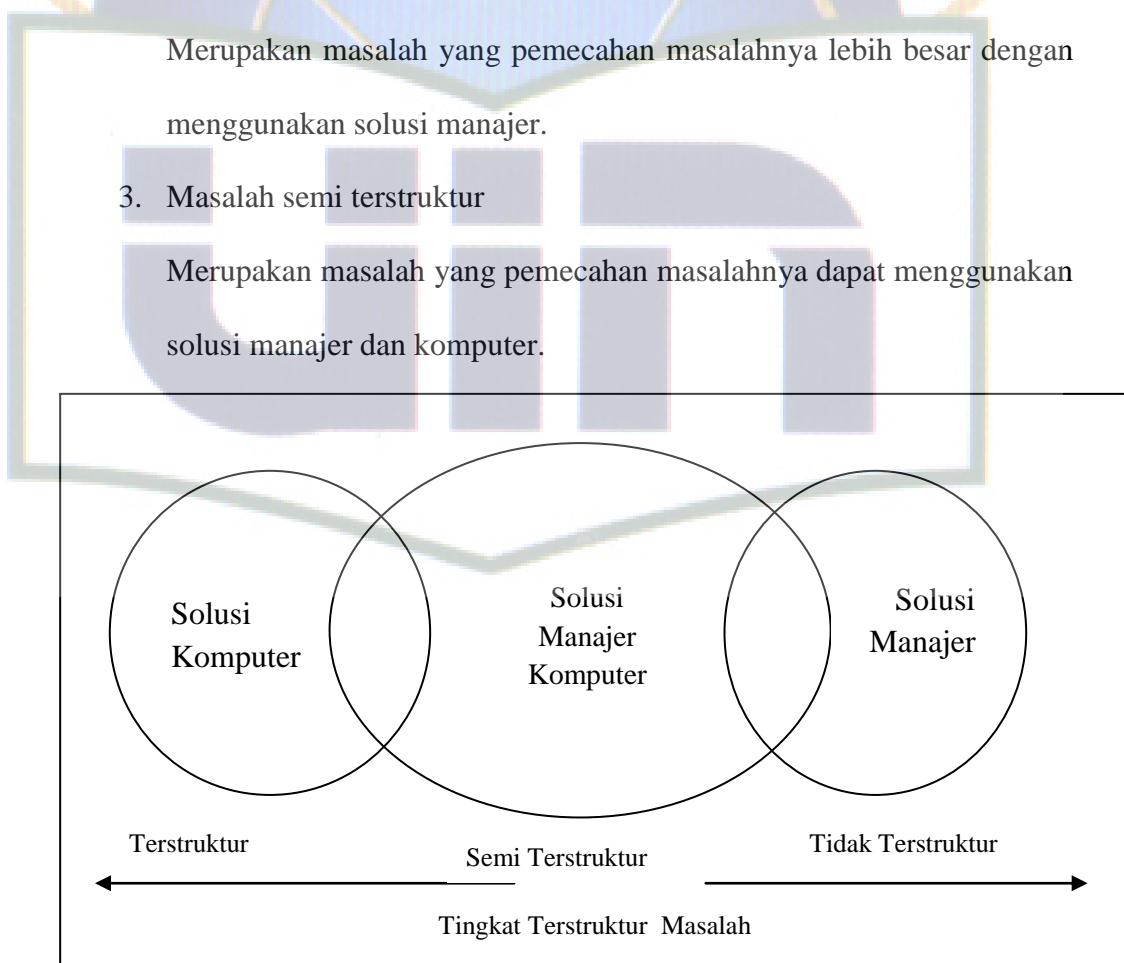
Masalah ini merupakan masalah yang pemecahannya lebih besar, dapat diselesaikan dengan menggunakan solusi komputer.

2. Masalah tidak terstruktur.

Merupakan masalah yang pemecahan masalahnya lebih besar dengan menggunakan solusi manajer.

3. Masalah semi terstruktur

Merupakan masalah yang pemecahan masalahnya dapat menggunakan solusi manajer dan komputer.



Gambar 2.1 SPK Berfokus pada Masalah-masalah semi terstruktur (McLeod, 2004)

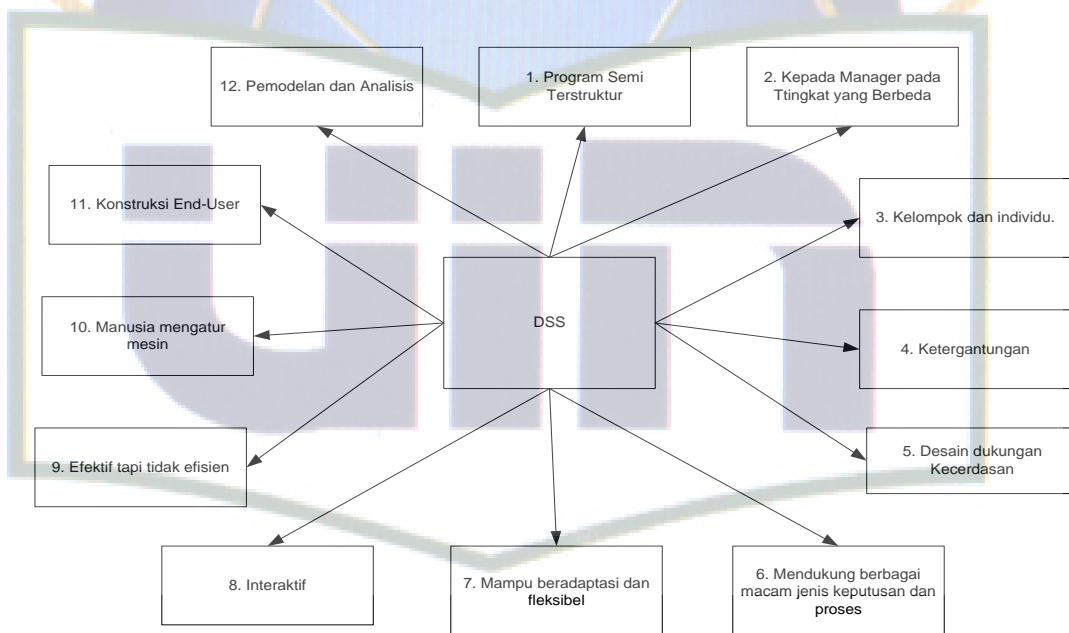
### 2.4.3 Dukungan Keputusan

SPK tidak dimaksudkan untuk menggantikan manajer. Komputer dapat diterapkan pada bagian masalah yang terstruktur, tapi manajer bertanggung jawab atas bagian yang tidak terstruktur, menerapkan penilaian atau intuisi dan melakukan analisis. Manajer dan komputer bekerja sama sebagai tim pemecahan masalah dalam memecahkan masalah yang ada di era semi terstruktur. (Turban, 2005)

### 2.4.4 Karakteristik dan Kemampuan Sistem Penunjang Keputusan.

1. Karakteristik sistem penunjang keputusan adalah:
    1. Merupakan sistem informasi berbasis komputer.
    2. Digunakan oleh para manajer (*knowledge worker*).
    3. Digunakan dalam pengambilan keputusan.
    4. Digunakan untuk membantu (mendukung) manajer, dan bukan menggantikan.
  5. Digunakan untuk keputusan yang semi terstruktur.
  6. Mendayagunakan model untuk menganalisis keputusan.
  7. Mendayagunakan *Database*.
2. Beberapa kemampuan sistem penunjang keputusan, yaitu:
    1. Mendukung dalam pengambilan keputusan untuk masalah semi terstruktur.
    2. Mudah digunakan.

3. Mendukung efektivitas pengambilan keputusan sehingga yang diambil lebih berkualitas.
4. Tersedia untuk para manajer dalam tingkat yang berbeda.
5. Mendukung gaya dan proses pengambilan keputusan.
6. Dapat digunakan oleh individu dan kelompok atau *group* dalam pengambilan keputusan.
7. Mendayagunakan suatu model untuk menganalisis suatu keputusan.
8. Mendukung semua tahap dalam proses pengambilan keputusan.



Gambar 2.2 Karakteristik dan Manfaat SPK (Turban, 2005)

#### 2.4.5 Pegambilan Keputusan

Pengambilan keputusan merupakan sebuah proses pemilihan diantara beberapa alternatif tindakan untuk mencapai tujuan (Turban, 2005).

Keputusan-keputusan dibuat untuk memecahkan masalah. Dalam usaha untuk memecahkan masalah pemecahan masalah mungkin membuat banyak keputusan. Keputusan merupakan rangkaian tindakan mengurangi dampak negatif, atau untuk memanfaatkan kesempatan (Mcleod,2004).

Sebelum memahami *modelling* (pemodelan), ada baiknya utnuk mengikuti suatu proses pengambilan keputusan sistematis, yang menurut simon, melibatkan tiga tahap utama: kecerdasan/*intelligence*, keputusan dan pilihan untuk tahap keempat adalah implementasi, telah ditambahkan, kemudian *monitoring* dapat dipertimbangkan menjadi tahap uang kelima.

Bagaimanapun, kita memandang *monitoring* seperti tahap kecerdasan/*intelegence* yang diberlakukan untuk tahap implementasi. Model Simon's adalah model yang paling singkat, namun model tersebut dapat melengkapi peranan pembuat keputusan secara rasional.

Ada arus aktivitas yang merupakan kelanjutan dari kecerdasan/*intelegence*, yang digambarkan dalam bentuk gambar (model). Arus tersebut menggunakan umpan balik yang akan mengarahkan langkah-langkah pengambilan keputusan. *Modelling* (pemodelan) adalah salah satu bagian penting dari proses yang secara alami akan menggambarkan tahap-tahap dari penemuan masalah sampai kepada solusinya untuk membantu dalam pengambilan keputusan melalui umpan balik.

### 1. *Intelligence*

Adalah ilmu yang dapat digunakan untuk mengidentifikasi masalah dan menggambarkan masalah tersebut menjadi suatu bentuk yang mudah dimengerti.

### 2. Design

Adalah suatu model yang menghadirkan sistem secara terstruktur. Ini dilakukan dengan pembuatan asumsi yang akan menyederhanakan masalah dengan mencatat hubungan antar semua variabel. Model tersebut akan dipertimbangkan, dan ukuran-ukuran akan diatur untuk dievaluasi supaya dapat mengenal berbagai macam tindakan alternatif yang dihasilkan. Sering sekali dalam proses konstruksi model ini akan mengidentifikasi potensial dari alternatif solusi.

### 3. Selection

Meliputi pemilihan suatu solusi dari model yang telah diusulkan tetapi bukan kepada masalah yang utama. Solusi ini akan diuji untuk menentukan kelangsungan model. Apabila solusi yang diusulkan layak untuk digunakan, kita akan siap untuk tahap yang terakhir.

### 4. Implementation

Implementasi yang sukses dapat menyelesaikan atau memecahkan masalah secara nyata. Kegagalan dari implementasi akan mengarahkan kita melalui *feedback* (umpan balik) kembali kepada tahap awal dari proses.

#### 2.4.6 Keuntungan Penggunaan Sistem Penunjang Keputusan

Penggunaan SPK dapat memberikan beberapa keuntungan, antara lain:

1. Kemampuan untuk mendukung terhadap pemecahan masalah-masalah yang kompleks.
2. Respon yang cepat terhadap suatu sistem yang tidak terduga karena hasil dari perubahan kondisi.
3. Meningkatkan pengendalian manajemen dan *performance* (prestasi).
4. Menghasilkan keputusan-keputusan yang subjektif dan lebih konsisten dari pada keputusan-keputusan yang dibuat dari intuisi.
5. Meningkatkan efektivitas manajerial.
6. Meningkatkan produktifitas analisa.
7. *Cost Saving*, penghematan biaya dapat terjadi karena menghindari keputusan yang salah.

## 2.5 Pengertian Bank Pembiayaan Rakyat Syariah (BPRS).

Bank Pembiayaan Rakyat Syariah yang biasa disebut BPRS adalah Bank Pembiayaan Rakyat Syariah sebagaimana dimaksud dalam Undang-Undang Nomor 21 Tahun 2008 tentang Perbankan Syariah. Bank Pembiayaan Rakyat Syariah adalah yang dalam kegiatannya tidak memberikan jasa dalam lalu lintas pembayaran.

Pembiayaan adalah penyediaan dan atau tagihan yang dipersamakan dengan itu berupa (amalia, 2007):

- a. Transaksi bagi hasil dalam bentuk mudharabah dan Musyarakah.
- b. Transaksi sewa menyewa dalam bentuk ijarah atau Sewa beli dalam bentuk *Ijarah Muntahiya Bit Tamlik*.

- c. Transaksi jual beli dalam bentuk piutang murabahah salam dan *istishna*
- d. Transaksi pinjam meminjam dalam bentuk piutang *qardh*.
- e. Transaksi sewa menyewa jasa dalam bentuk *ijarah*. Untuk transaksi multijasa.

Berdasarkan persetujuan atau kesepakatan antara bank Syariah dan atau UUS dan pihak lain yang mewajibkan pihak yang dibiayai dan atau diberi fasilitas dana untuk mengembalikan dana tersebut setelah jangka waktu tertentu dengan imbalan *ujrah* tanpa imbalan atau bagi hasil.

## 2.6 Konsep Pengambilan Keputusan Pembiayaan UKM.

### 2.6.1 Pengertian Usaha Kecil Menengah (UKM).

Yang dimaksud dengan Usaha Kecil menurut Undang-Undang No. 9 tahun 1995 pasal 5 ialah kegiatan ekonomi rakyat yang berskala kecil dan memenuhi kriteria sebagai berikut:

Memiliki kekayaan bersih paling banyak Rp. 200.000.000,- (Dua Ratus Juta Rupiah) tidak termasuk tanah dan bangunan tempat usaha atau Memiliki hasil penjualan tahunan paling banyak Rp. 1.000.000.000,- (Satu Miliar Rupiah)

1. Milik Warga Negara Indonesia
2. Berdiri sendiri, bukan merupakan anak perusahaan atau cabang perusahaan yang tidak dimiliki, dikuasai, atau berafiliasi baik langsung maupun tidak langsung dengan Usaha Menengah atau Usaha Besar.

3. Berbentuk usaha orang perorangan, badan usaha yang tidak berbadan hukum, atau badan usaha yang berbadan hukum, termasuk koperasi.

Sementara menurut Inpres No.10/1999, yang dimaksud dengan usaha menengah adalah intensitas usaha dengan asset bersih Rp.200 juta – Rp. 10 Miliar termasuk tanah dan bangunan.

Usaha Kecil adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria Usaha Kecil sebagaimana dimaksud dalam Undang-Undang ini.

Usaha Menengah adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perseorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan usaha kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sebagaimana diatur dalam Undang-Undang.

Tabel 2.1 Perbandingan Nilai Uang Dalam UKM

<b>No</b>	<b>Uraian</b>	<b>Kriteria</b>	
		<b>Asset</b>	<b>Omzet</b>
1	Usaha Mikro	Max 50 juta	Max 300 juta
2	Usaha Kecil	> 50 juta – 500 juta	> 300 juta - 2,5 Milyar
3	Usaha Menengah	> 500 Juta – 10 Milyar	> 2,5 Milyar – 50 Milyar

## 2.6.2 Pengertian Pembiayaan.

Pembiayaan yaitu pendanaan yang diberikan oleh suatu pihak kepada pihak lain untuk mendukung investasi yang telah direncanakan, baik dilakukan sendiri maupun lembaga. Dengan kata lain, pembiayaan adalah pendanaan yang dikeluarkan untuk mendukung investasi yang telah direncanakan. (Muhammad,2005)

Pembiayaan adalah penyelesaian atau uang tagihan yang dapat dipersamakan dengan berdasarkan persetujuan antara bank dengan pihak yang mewajibkan pihak yang dibiayai untuk mengembalikan uang atau tagihan tersebut setelah jangka waktu tertentu dengan pembagian hasil keuntungan.

Jadi yang dimaksud pembiayaan yaitu pendanaan yang diberikan oleh satu pihak kepada pihak lain untuk mendukung investasi yang telah direncanakan, baik dilakukan sendiri maupun lembaga dengan kata lain pembiayaan bisa juga sebagai pendanaan yang dikeluarkan untuk mendukung investasi yang telah direncanakan.

### 1. Dalil Syariah

- Al-Qur'an Surat Al-Muzzamil ayat 20*

وَعَلِمَ أَنَّ سَيْكُونُ مِنْكُمْ مَرْضَىٰ وَأَخْرُونَ يَصْرِفُونَ فِي الْأَرْضِ يَتَغَوَّنُونَ مِنْ فَضْلِ اللَّهِ

وَأَخْرُونَ يُقْتَلُونَ فِي سَبِيلِ اللَّهِ فَاقْرَءُوا مَا تَيَسَّرَ مِنْهُ وَأَقِيمُوا الصَّلَاةَ وَأَطْوِلُوا الزَّكَوَةَ

وَأَقْرِضُوا اللَّهَ قَرْضًا حَسَنًاٰ وَمَا تُقْدِمُوا لِأَنفُسِكُمْ مِنْ خَيْرٍ تَحْدُوهُ عِنْدَ اللَّهِ هُوَ خَيْرٌ

وَأَعْظَمُ أَجْرًا وَآسْتَغْفِرُوا اللَّهَ إِنَّ اللَّهَ غَفُورٌ رَّحِيمٌ ﴿١٣﴾

“...Dia mengetahui bahwa akan ada di antara kamu orang-orang yang sakit dan orang-orang yang berjalan di muka bumi mencari sebagian karunia Allah dan orang-orang yang lain lagi berperang di jalan Allah. Maka bacalah apa yang mudah (bagimu) dari Al Quran dan dirikanlah sembahyang, tunaikanlah zakat dan berikanlah pinjaman kepada Allah pinjaman yang baik. Dan kebaikan apa saja yang kamu perbuat untuk dirimu niscaya kamu memperoleh (balasannya) di sisi Allah sebagai balasan yang paling baik dan yang paling besar pahalanya. Dan mohonlah ampunan kepada Allah. Sesungguhnya Allah Maha Pengampun lagi Maha Penyayang”

b. *Al-Quran Surat An-Nisa ayat 29*

يَأَيُّهَا الَّذِينَ إِيمَنُوا لَا تَأْكُلُوا أَمْوَالَكُمْ بَيْنَكُمْ بِالْبَطْلِ إِلَّا أَنْ تَكُونَ حِلْةً

عَنْ تَرَاضٍ مِّنْكُمْ وَلَا تَقْتُلُوا أَنفُسَكُمْ إِنَّ اللَّهَ كَانَ بِكُمْ رَّحِيمًا ﴿٢٩﴾

29. Hai orang-orang yang beriman, janganlah kamu saling memakan harta sesamamu dengan jalan yang batil, kecuali dengan jalan perniagaan yang berlaku dengan suka sama-suka di antara kamu. Dan janganlah kamu membunuh dirimu; sesungguhnya Allah adalah Maha Penyayang kepadamu.

### c. Dalil Hadist

Diriwayatkan oleh Ibnu Abbas bahwasanya, Sayidina Abbas jikalaui memberikan dana kemitraan usahanya secara *Mudharabah*, ia mensyaratkan agar dananya tidak dibawa mengarungi lautan, menuruni lembah yang berbahaya menyalahi peraturan maka yang bersangkutan bertanggung jawab atas dana tersebut. Disampaikanlah syarat-syarat tersebut kepada Rasulullah SAW dan Rasulpun memperkenankannya

HR. Abu Daud, Baihaqi dan Al Hakam: “*Rahmat Allah SWT tercurahkan atas dua pihak yang sedang bekerja sama selama mereka tidak melakukan penghianatan, manakala berkhianat maka bisnisnya akan tercela dan keberkahan pun akan sirna daripadanya*”

#### 2.6.2.1 Jenis-Jenis Pembiayaan

Jenis pembiayaan dapat dikelompokkan kedalam beberapa aspek, diantaranya (Muhammad,2005):

##### a. Pembiayaan menurut Produktif

Ditunjukan untuk meningkatkan kebutuhan produksi secara luas, baik usaha produk, perdagangan, maupun investasi.

##### b. Pembiayaan menurut Konsumtif

Digunakan untuk memenuhi kebutuhan konsumsi yang akan dihabiskan digunakan untuk memenuhi kebutuhan.

### 2.6.2.2 Pembiayaan Menurut Jangka Waktu.

- a. Pembiayaan jangka waktu pendek, yaitu pembiayaan yang dilakukan dalam waktu 1 bulan sampai dengan 1 tahun.
- b. Pembiayaan jangka waktu menengah, yaitu pembiayaan yang dilakukan dalam waktu 1 tahun sampai dengan 5 tahun.
- c. Pembiayaan jangka waktu panjang, yaitu pembiayaan yang dilakukan dalam waktu lebih dari 5 tahun (Muhammad,2005).

### 2.6.2.3 Pembiayaan Menurut Segmentasi Pasar.

- a. Pembiayaan usaha, yaitu pembiayaan yang diberikan kepada pedagang atau pengusaha untuk perputaran maupun prasarana dan sarana usaha, contoh pembiayaan kepada pedagang bakso, sembako.
- b. Pembiayaan kolektif, yaitu pembiayaan yang diberikan kepada kelompok karyawan yang bernaung dibawah suatu lembaga seperti yayasan (Muhammad,2005).

## 2.6.3 Tujuan dan Fungsi Pembiayaan

Tujuan Pembiayaan antara lain:

- a. Memperoleh bagi hasil dari modal yang disimpan serta memperoleh kesejahteraan dari bank yang dikelolanya.
- b. Membantu mengembangkan usaha.
- c. Memperoleh barang yang dibutuhkan.
- d. Mengurangi pengangguran.

- e. Dapat meneruskan dan mengembangkan usaha agar tetap survival dan memperluas jaringan usaha, sehingga makin banyak masyarakat yang dapat terlayani.

Fungsi pembiayaan antara lain:

- a. Meningkatkan daya guna uang dan barang.
- b. Meningkatkan peredaran uang.
- c. Menjaga stabilitas ekonomi.
- d. Meningkatkan pendapatan nasional.
- e. Penghubung ekonomi internasional
- f. Menimbulkan kegairahan berusaha dan memperlancar produksi serta konsumsi sehingga taraf hidup masyarakat meningkat

(Muhammad,2005).

#### **2.6.4 Prosedur Dalam Pemberian Pembiayaan.**

Prosedur pemberian dan penilaian pembiayaan oleh dunia perbankan secara umum antar bank yang satu dengan bank yang lain tidak jauh berbeda. Yang menjadi perbedaan mungkin hanya terletak dan prosedur dan persyaratan yang diterapkan dengan pertimbangan masing-masing.

Dengan memperhatikan ketentuan umum manajemen pembiayaan di bank syariah. Ada beberapa aspek yang perlu diperhatikan dalam prosedur analisis pembiayaan. Aspek-aspek penting dalam anisis

pembiayaan yang perlu dipahami oleh pengelola bank syari'ah (Muhammad,2005).

Prosedur analisis :

1. Berkas dan pencatatan.
2. Data pokok dan analisis pendahuluan.
  - a. Realisasi pembelian, produksi dan penjualan.
  - b. Rencana pembelian, produksi dan penjualan.
  - c. Jaminan.
  - d. Laporan keuangan.
- e. Data kualitatif dari calon debitur.
3. Penelitian data.
4. Penelitian data realisasi usaha.
5. Penelitian dan penilaian barang jaminan.
6. Laporan keuangan dan penelitiapnnya.

**Keputusan Permohonan Pembiayaan:**

1. Bahan pertimbangan pengambilan keputusan.
2. Wewenang pengambilan keputusan.

### **2.6.5 Prinsip-Prinsip Pemberian Pembiayaan.**

Sebelum fasilitas pembiayaan diberikan maka bank harus merasa yakin bahwa pembiayaan yang diberikan benar-benar akan kembali. Keyakinan tersebut diperoleh dari penilaian pembiayaan sebelum pembiayaan tersebut disalurkan. Penilaian pembiayaan oleh bank dapat

dilakukan dengan berbagai cara untuk mendapatkan keyakinan tentang nasabahnya seperti melalui prosedur penilaian yang benar.

Kriteria penilaian yang harus dilakukan oleh bank untuk mendapatkan nasabah yang benar-benar menguntungkan dilakukan dengan analisis 5 C dan 7 P Adapun penjelasan analisis dengan 5C adalah sebagai berikut (Kasmir,2005):

#### 1. *Character*

Suatu keyakinan bahwa, sifat atau watak dari orang-orang yang akan diberikan pembiayaan benar-benar dapat dipercaya yang tercermin dari latar belakang nasabah. Baik latar belakang pekerjaan maupun yang bersifat pribadi seperti cara hidup atau gaya hidup yang dianutnya, keadaan keluarga, hobi dan sosial standingnya.

#### 2. *Capacity*

Untuk melihat nasabah dalam kemampuannya dalam bidang bisnis yang dihubungkan dengan pendidikannya, begitu pula kemampuannya dalam menjalankan usahanya selama ini.

#### 3. *Capital*

Untuk melihat penggunaan modal apakah efektif, dilihat dari laporan keuangan (neraca laporan rugi laba). *Capital* juga dilihat dari sumber mana saja modal yang ada sekarang.

#### 4. *Colleteral*

Merupakan jaminan yang diberikan calon nasabah baik yang bersifat fisik maupun *non* fisik. Jaminan hendaknya melebihi jumlah

pembiayaan yang diberikan.

##### 5. *Condition*

Dalam menilai pembiayaan hendaknya juga dinilai kondisi ekonomi dan politik sekarang dan dimasa yang akan datang sesuai sektor masing-masing, serta prospek usaha dari sektor yang ia jalankan (Muhammad,2005).

Kemudian penilaian pembiayaan dengan metode analisis 7 P adalah sebagai berikut :

###### 1. *Personality*

Yaitu menilai nasabah dari segi kepribadiannya atau tingkah lakunya sehari-hari maupun masa lalunya.

###### 2. *Part*

Yaitu mengklasifikasikan nasabah ke dalam klasifikasi tertentu atau golongan-golongan tertentu berdasarkan modal, loyalitas serta karakternya.

###### 3. *Purpose*

Yaitu untuk mengetahui tujuan nasabah dalam mengambil pembiayaan, termasuk jenis pembiayaan yang diinginkan nasabah.

###### 4. *Prospect*

Yaitu untuk menilai nasabah dimasa yang akan datang menguntungkan atau tidak.

###### 5. *Payment*

Merupakan ukuran bagaimana cara nasabah mengembalikan pemberian yang telah diambil atau dari sumber mana saja dana untuk pengembalian pemberian.

#### 6. *Profitability*

Untuk menganalisis bagaimana kemampuan nasabah dalam mencari laba. *Profitability* diukur dan periode ke periode apakah akan tetap sama atau akan semakin meningkat.

#### 7. *Protection*

Tujuannya adalah bagaimana menjaga agar usaha dan jaminan mendapatkan perlindungan. Perlindungan dapat berupa jaminan barang atau jaminan asuransi.

### 2.7 *Artificial Intelligence.*

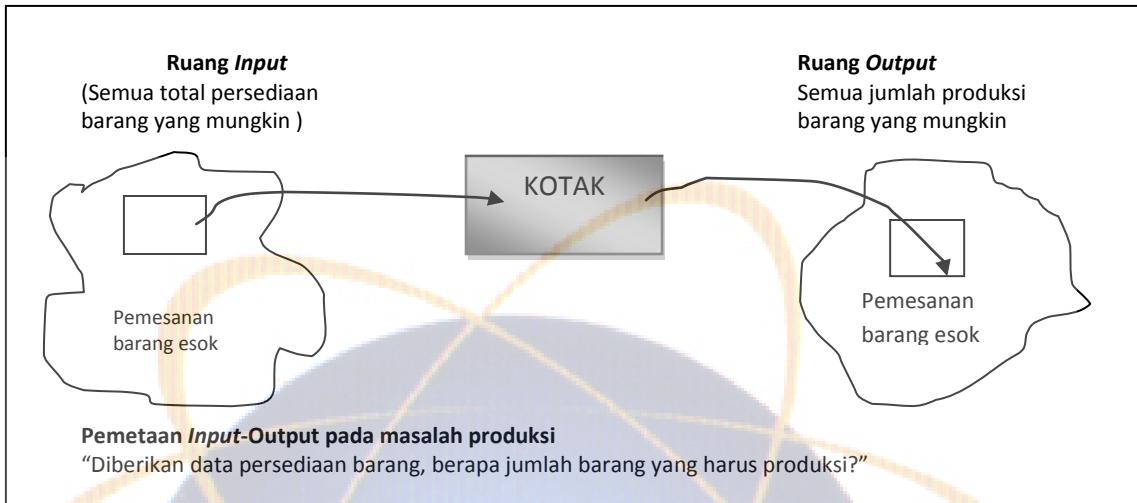
Sistem pakar adalah sistem informasi berbasis komputer yang menggunakan pengetahuan pakar untuk mencapai performa keputusan tingkat tinggi dalam *domain* persoalan yang sempit (Nedovic dalam Turban, 2005).

Karakteristik ‘cerdas’ sudah mulai dibutuhkan diberbagai disiplin ilmu dan teknologi mengklasifikasikan lingkup kecerdasan buatan tersebut berdasarkan pada *output* yang diberikan, yaitu pada aplikasi komersil (meskipun kecerdasan buatan itu sendiri bukan merupakan medan komersil) Lingkup utama dalam kecerdasan buatan tersebut adalah (Kusumadewi,2004).:

- a. Sistem Pakar (*Expert System*) dimana komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar.
- b. Pengolahan bahasa alami (*Natural Language Processing*), dimana komputer dapat berkomunikasi dengan user dengan bahasa sehari-hari.
- c. Pengenalan Ucapan (*Speech Recognition*) dimana komputer dapat mengenali suara sehingga dapat berkomunikasi dengan manusia.
- d. Robotika dan Sistem Sensor (*Robotika dan Sensory System*)
- e. *Computer Vision*, dimana komputer dapat menginterpretasikan gambar atau *object-object* tampak.
- f. *Intelegent Computer- Aided Instruction* dimana komputer digunakan sebagai tutor yang dapat melatih dan mengajar.
- g. Permainan Komputer (*Game Playing*)
  - Seiring dengan perkembangan teknologi, muncul beberapa teknologi yang bertujuan untuk membuat agar komputer menjadi cerdas sehingga menirukan kerja manusia sehari-hari. Teknologi ini juga mampu mengakomodasi adanya ketidakpastian data *input*, diantara logika *Fuzzy* dan algoritma genetika.

## 2.8 Logika *Fuzzy*.

Menurut (Kusumadewi,2004), Logika *Fuzzy* adalah suatu cara yang tepat memetakan suatu ruangan *input* ke dalam suatu ruang *output*. Contoh pemetaan suatu *input-output* dalam bentuk grafis terlihat pada gambar 2.3



Gambar 2.3 contoh Pemetaan *Input-Output*

Logika *Fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *Fuzzy* modern dan metode baru ditemukan beberapa tahun lalu, namun sebenarnya konsep tentang logika *Fuzzy* telah ada pada diri manusia sejak lama. Logika *Fuzzy* merupakan bagian dari logika boolean, yang digunakan untuk menangani konsep derajat kebenaran, yaitu nilai kebenaran antara benar dan salah. Teori ini diperkenalkan oleh Dr. Lotfi Zadeh dari Barkeley pada tahun 1960-an, sebagai model ketidakpastian.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *Fuzzy*. Yaitu:

1. Variabel *Fuzzy*

Variabel *Fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *Fuzzy* contoh umur, temperatur, permintaan dan sebagainya.

2. Himpunan *Fuzzy*

Himpunan *Fuzzy* adalah himpunan dengan *Fuzzy* keanggotaan menunjukkan bahwa suatu sistem alam semesta pembicaraan tidak hanya berada pada 0 atau 1 namun nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran suatu item tidak hanya bernilai benar (1) atau salah (0) namun masih ada nilai-nilai yang terletak antara benar dan salah (Kusumadewi,2004).

Himpunan *Fuzzy* merupakan suatu *group* yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *Fuzzy*. Suatu himpunan *Fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan  $\mu_A$ , yang harganya berada dalam interval [0,1].

Secara matematis dinyatakan dengan :

$$\mu_A: U \rightarrow [0,1]$$

himpunan *Fuzzy* A dalam semesta pembicaraan U biasanya dinyatakan sebagai kumpulan pasangan elemen  $(u, \mu_A(u))$  ( $u$  anggota U) dan besarnya derajat keanggotaan elemen tersebut dinyatakan dengan:

$$A = \{(u, \mu_A(u)) / u \in U\}$$

Contoh: variabel umur terbagi menjadi tiga himpunan *Fuzzy*, yaitu muda, parobaya, dan tua.

Himpunan *Fuzzy* memiliki dua *attribute*, yaitu:

- *Linguistik*; yaitu penamaan suatu *group* yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti muda, parobaya, tua.

- *Numeris*; yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 40,25,50.

### 3. Semesta Pembicaraan

Semesta Pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *Fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak ada batas atasnya. Contoh: semesta pembicaraan untuk variabel umur adalah 0 sampai ~.

### 4. Domain

Domain himpunan *Fuzzy* adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *Fuzzy*. *Domain* merupakan himpunan bilangan real yang senantiasa (bertambah) secara monoton. Nilai semesta pembicaraan dapat berupa bilangan positif.

Contoh *Domain Fuzzy* pada umur.

$$\begin{array}{ll} \text{Muda} & = [0 \quad 45] \\ \text{Parobaya} & = [35 \quad 55] \\ \text{Tua} & = [45 \quad \sim] \end{array}$$

### 5. *Fuzzifikasi*

*Fuzzifikasi* adalah proses untuk mengubah variabel-variabel *non Fuzzy* (variabel numerik) menjadi variabel *Fuzzy* (variabel linguistik) nilai masukan-masukan yang masih dalam bentuk variabel numerik

sebelum diolah oleh pengendali logika *Fuzzy* harus diubah terlebih dahulu kedalam variabel *Fuzzy*. Melalui fungsi keanggotaan yang telah disusun maka nilai-nilai masukan tersebut menjadi reformasi *Fuzzy* yang berguna nantinya untuk proses pengolahan.

#### 6. *Defuzzifikasi*

Dalam logika *Fuzzy*, hubungan antara masukan dan keluaran secara umum dinyatakan dengan:

*IF (A<sub>1</sub>) THEN (B<sub>1</sub>)*

.....

*IF (A<sub>n</sub>) THEN (B<sub>n</sub>)*

A<sub>1</sub>... A<sub>n</sub> adalah *antecedent*, yaitu masukan yang telah *defuzzifikasi* kan, sedangkan B<sub>1</sub>...B<sub>n</sub> adalah *consequent*, yaitu aksi pengendalian (keluaran). Hubungan antara *antecedent* dan *consequent*

Proses untuk mendapat aksi keluaran dari suatu kondisi masukan dengan mengikuti *rule-rule* yang telah dibuat disebut *inference reasoning* (pengambilan keputusan).

Keputusan yang dihasilkan dari proses penalaran masih dalam bentuk *Fuzzy*, yaitu berupa derajat keanggotaan keluaran. Hasil ini harus diubah kembali menjadi variabel numerik *non-Fuzzy* melalui proses *defuzzifikasi*.

#### 2.8.1 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai

keanggotaannya (sering disebut derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

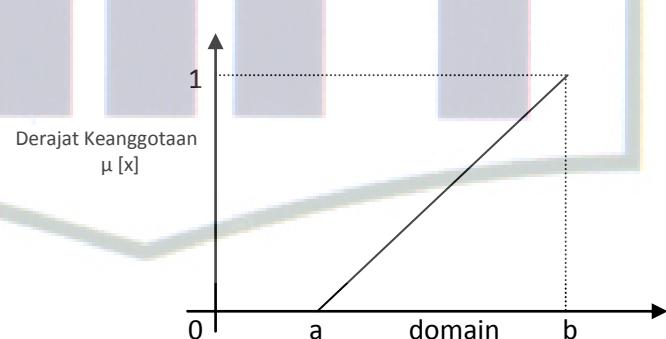
Ada beberapa fungsi yang digunakan (Kusumadewi,2004).:

### 1. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada dua keadaan himpunan *Fuzzy* linear, yaitu:

1. Kenaikan himpunan dimulai pada nilai *domain* yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai *domain* yang memiliki derajat keanggotaan lebih tinggi.

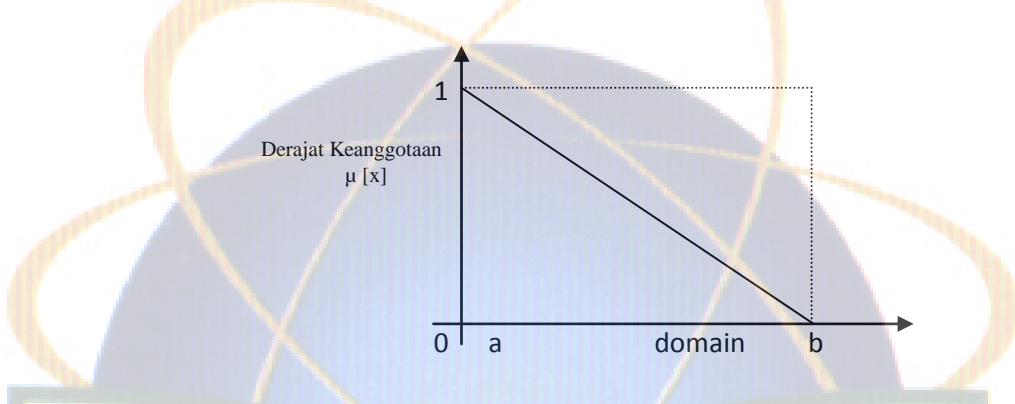


Gambar 2.4 Representasi Linear Naik

### Fungsi Keanggotaan

$$\mu [x] = \begin{cases} 0; & x \leq a \\ (x - a) / (b - a); & a \leq x \leq b \\ 1 & x \geq b \end{cases}$$

2. Garis lurus dimulai dari nilai *domain* dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai *domain* yang memiliki derajat keanggotaan lebih rendah.



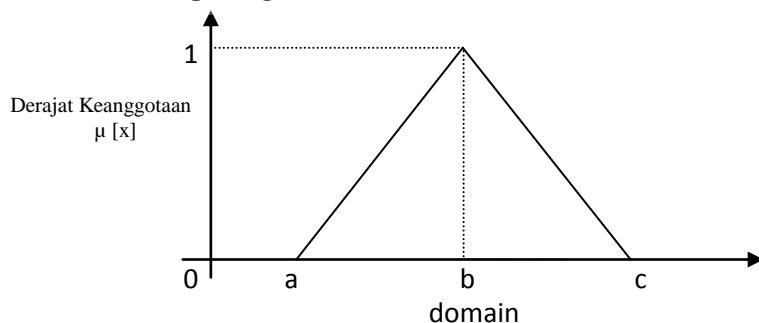
Gambar 2.5 Representasi Linear Turun

Fungsi Keanggotaan

$$\mu [x] = \begin{cases} (b - x) / (b - a); & a \leq x \leq b \\ 0; & x \geq b \end{cases}$$

### 2.8.2 Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis (linear) terlihat pada gambar 2.6

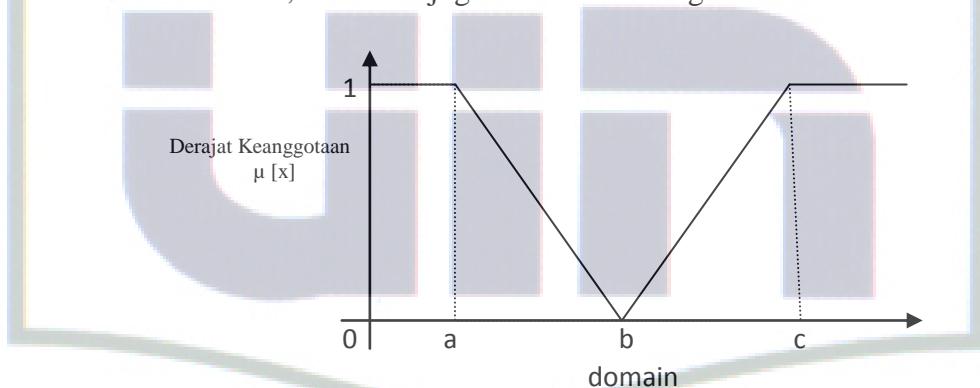


Gambar 2.6 Representasi Kurva Segitiga

$$\mu [x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a) / (b - a); & a \leq x \leq b \\ (b - x) / (c - b); & b \leq x \leq c \end{cases}$$

### 2.8.3 Representasi Kurva Bentuk Bahu

Daerah yang terletak ditengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik-turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himpunan Fuzzy ‘bahu’ bukan segitiga digunakan untuk mengakhiri variabel suatu daerah Fuzzy. Bahu kiri bergerak dari benar kesalah, demikian juga bahu kanan bergerak dari salah ke benar.



Gambar 2.7 Representasi Kurva Bahu Fungsi keanggotaan kurva bahu

sama dengan representasi linear

### 2.9 Fuzzy Inference System.

Menurut (Kusumadewi, 2004) ada beberapa metode yang digunakan dalam *Fuzzy Inference System*.

### 2.9.1 Metode Tsukamoto

Pada metode Tsukamoto, setiap konsekuensi pada aturan yang berbentuk *IF-THEN* harus direpresentasikan dengan suatu himpunan *Fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil *inferensi* dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Misalkan ada variabel input, yaitu  $x$  dan  $y$ , serta satu variabel output yaitu  $z$ . Variabel  $x$  terbagi atas 2 himpunan yaitu  $A_1$  dan  $A_2$ , variabel  $y$  terbagi atas 2 himpunan juga, yaitu  $B_1$  dan  $B_2$ , sedangkan variabel output  $Z$  terbagi atas 2 himpunan yaitu  $C_1$  dan  $C_2$ . Tentu saja himpunan  $C_1$  dan  $C_2$  harus merupakan himpunan yang bersifat monoton. Diberikan 2 aturan sebagai berikut:

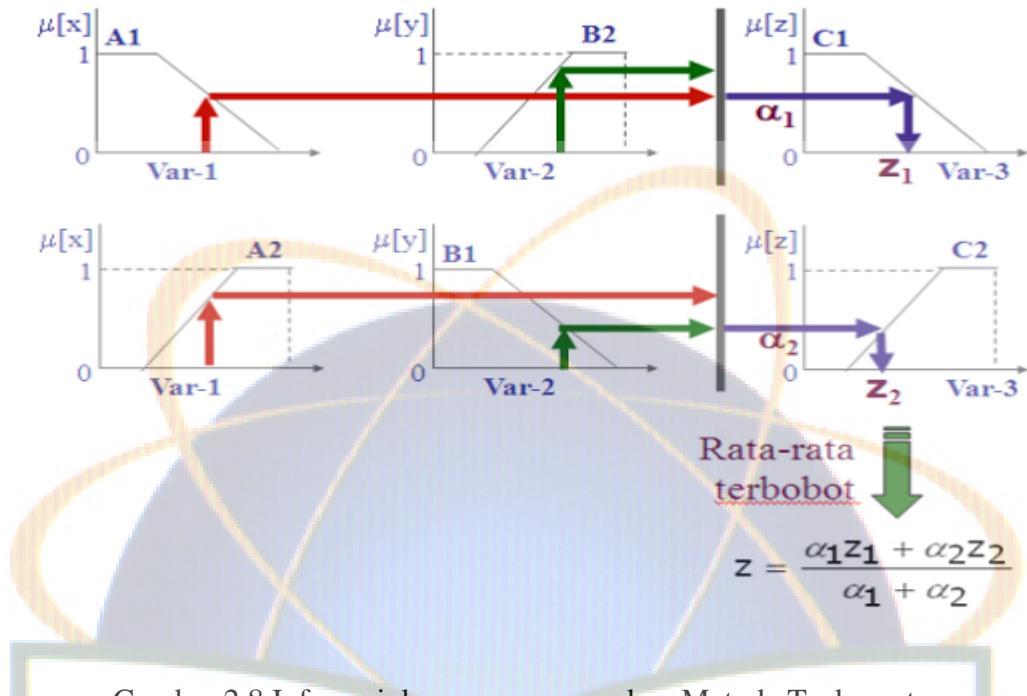
IF  $x$  is  $A_1$  and  $y$  is  $B_2$  THEN  $z$  is  $C_1$

IF  $x$  is  $A_2$  and  $y$  is  $B_2$  THEN  $z$  is  $C_1$

@predikat untuk aturan pertama dan kedua, masing-masing adalah  $a_1$  dan  $a_2$  dengan menggunakan penalaran monoton, diperoleh nilai  $Z_1$  pada aturan pertama, dan  $Z_2$  pada aturan kedua. Terakhir dengan menggunakan aturan terbobot, diperoleh hasil akhir dengan formula sebagai berikut:

$$Z = \frac{a_1 Z_1 + a_2 Z_2}{a_1 + a_2}$$

Diagram blok proses inferensi dengan metode tsukamoto (Jang, 1997) dapat dilihat pada Gambar 2.8.



Gambar 2.8 Inferensi dengan menggunakan Metode Tsukamoto

### 2.9.2 Metode Mamdani

Metode ini sering juga dikenal dengan metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Dalam metode Mamdani diperlukan empat tahapan, yaitu pembentukan himpunan *Fuzzy*, aplikasi fungsi implikasi aturan (*rule*), komposisi aturan, penegasan (*Defuzzify*).

### 2.9.3 Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* (konsekuensi) sistem tidak berupa himpunan *Fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Dilihat dari *output* yang dihasilkan dari ketiga metode tersebut, metode Tsukamoto dapat menghasilkan *output* yang lebih besar dibandingkan metode yang lainnya. Untuk lebih jelasnya lagi perhatikan tabel studi kasus dibawah ini:

Data 1 bulan Terakhir Produksi Makanan Kaleng Jenis ABC

Tabel 2.2 Produksi makanan kaleng jenis ABC

Jumlah	Permintaan (Kemasan/hari)	Persediaan (Kemasan/hari)	Produksi (Kemasan/hari)
Max	5000	600	7000
Min	1000	100	2000

Dari tabel 2.2 tersebut, berupa kemasan makanan jenis ABC yang harus diproduksi, jika jumlah permintaan sebesar Rp.4.000,00 kemasan dan persediaan sebesar 300 kemasan.

Tabel 2.3 *Output* jumlah produksi dengan 3 metode Fuzzy

No	Permintaan	Persediaan	Jumlah Produksi		
			Tsukamoto	Mamdani	Sugeno
1	4000	300	4983	4248	4230

Pada tabel 2.3 dapat dilihat bahwa jumlah makanan kaleng jenis ABC yang harus diproduksi jika dihitung dengan menggunakan metode Tsukamoto sebesar 4.983, metode Mamdani sebesar 4248 dan metode Sugeno sebesar 4230 kemasan. Dari ketiga metode tersebut, metode Tsukamoto mampu menghasilkan *output* terbesar.

Jika dilihat pada data satu bulan terakhir perusahaan tersebut memproduksi makanan kaleng jenis ABC, metode Tsukamoto mampu

menutupi apabila jumlah maksimal yaitu 5000 kemasan-hari (lihat tabel 2.2). Sedangkan dua metode yang lain yaitu Mamdani dan Sugeno tidak dapat menutupi jika jumlah permintaan pasar maksimal.

Hal ini sangatlah cocok untuk mengantisipasi jika permintaan pasar mencapai jumlah maksimal, mengingat jumlah permintaan pasar yang tidak dapat diprediksi secara cepat.

Selain itu, hal ini juga cocok untuk efisiensi mesin dan SDM tiap hari, jika perusahaan akan menetapkan jumlah barang yang harus diproduksi esok hari dengan tetap mempertimbangkan jumlah produksi terakhir perusahaan tersebut dalam memproduksi makanan kaleng tiap harinya.

Oleh karena itu pada penelitian ini penulis menggunakan metode Tsukamoto dalam membangun sistem penilaian ini. Hal ini dikarenakan metode Tsukamoto memberikan *output* yang lebih besar, tetapi dalam kisaran yang wajar. Dengan *output* yang makin besar memungkinkan peluang pemberian pembiayaan kepada nasabah makin besar pula. Hal ini lebih baik karena sifat BPRS yang lebih fleksibel dalam memberikan pembiayaan, namun tetap memperhitungkan ketelitian, mengingat dalam kenyataannya banyak pengusaha kecil yang tidak layak masuk daftar mendapat layanan pembiayaan bank.

## 2.10 Tools Perancangan Sistem

### 2.10.1 Unified Modelling Language (UML)

*Unified Modelling Language (UML)* adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem berorientasi *object*, hal ini disebabkan karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan sharing dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005).

*UML* merupakan suatu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan *object* (Whitten, 2004).

### 2.10.2 Konsep Untuk Pemodelan *Object*

Berikut ini mengenai konsep sistem dalam pemodelan berorientasi *object* (Whitten, 2004):

#### 1. *Object*

Merupakan sesuatu yang ada atau dapat dilihat, disentuh, atau dirasakan dan *user* menyimpan data serta mencatat perilaku mengenai sesuatu itu.

#### 2. *Attribute*

Adalah data yang mewakili karakteristik *interest* tentang sebuah *object*.

### 3. *Encapsulation*

Adalah pengemasan beberapa item kedalam suatu unit. *Encapsulation* diterapkan pada suatu *object*, attribut dan *behavior object* dipaketkan bersama-sama.

### 4. *Behavior*

Merupakan kumpulan dari sesuatu yang dapat dilakukan oleh *object* dan terkait dengan fungsi-fungsi yang bertindak pada data *object* atau *attribute*.

### 5. *Class*

Merupakan satu set *object* yang memiliki attribut dan *behavior* yang sama.

## 2.10.3 Tipe Diagram UML

Beberapa diagram yang digunakan dalam menyusun skripsi ini adalah sebagai berikut (Whitten. 2004):

### 1. *Use Case Model Diagram*

*Use Case* diagram adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan

untuk berinteraksi dengan sistem. Dalam *Use Case* diagram memiliki pemodelan sebagai berikut:

a. *Use Case*

Pemodelan *Use Case* mengidentifikasi dan menggambarkan fungsi-fungsi sistem dari sudut pandang pengguna eksternal dan dalam sebuah cara dan terminologi yang mereka pahami. *Use Case* merupakan urutan langkah-langkah yang secara tindakan saling terkait (*scenario*), baik otomatis ataupun manual.

b. *Actor*

*Actor* merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. *Actor* dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun.

c. *Relationship*

Pada diagram *Use Case*, *relationship* digambarkan sebagai sebuah garis antara dua simbol.

## 2. *Activity Diagram*

*Activity diagram* secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis atau *Use Case*. Diagram ini berbeda dengan *flowchart* dimana diagram ini menyediakan sebuah mekanisme untuk menggambarkan kegiatan yang tampak secara paralel.

## 3. *Class Diagram*

*Class Diagram* menggambarkan struktur *object system*. Diagram ini menunjukkan *class object* tersebut. Dalam diagram ini terdapat *associations* dan *multiplicity*, *generalization / specialization relationship* dan *aggregation relationship*.

#### 4. Sequence Diagram.

*Sequence Diagram* secara grafis menggambarkan bagaimana *object* berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *Use Case* atau operasi. Diagram mengilustrasikan bagaimana pesan terkirim dan diterima diantara *object* dan *sequence* apa.

Diagram ini menunjukkan aliran fungsionalitas dalam *Use Case*.

#### 5. Statechart Diagram

*Statechart Diagram* digunakan untuk memodelkan *object behavior* khususnya yang dinamis. Diagram ini mengilustrasikan siklus hidup *object* dan keadaan yang dapat diasumsikan oleh *object* dan *event* yang menyebabkan *object*, misalkan sebuah *account* bank dapat eksis dalam beberapa keadaan yang berbeda. Seperti dapat buka, tutup atau kondisi *overdraw* ( kondisi dimana jumlah pengambilan lebih besar dari simpanan yang ada).

## 2.11 Bahasa Pemprograman

### 2.11.1 Java

*Java* diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari *Sun Microsystem*

yang memiliki kode *Green* dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitektur tertentu. Mulanya disebut OAK, tetapi karena OAK sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada, maka Sun mengubahnya menjadi *Java*.

Sun kemudian meluncurkan *browser* dari *Java* yang disebut *Hot Java* yang mampu menjalankan *Applet*. Setelah itu teknologi *Java* diadopsi oleh *Netscape* yang memungkinkan program *Java* dijalankan di *browser* *Netscape* yang kemudian diikuti *Internet Explorer*. Karena keunikannya dan kelebihannya, teknologi *Java* mulai menarik banyak vendor seperti *IBM*, *Symantec*, dan *Inprise*.

Sun merilis versi awal *Java* secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul *JDK 1.1*, kemudian *JDK 1.2* yang mulai disebut sebagai versi *Java2* karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah adanya *Swing* yang merupakan teknologi *GUI (Graphical User Interface)* yang mampu menghasilkan *window* yang *portabel*. Pada tahun 1998 – 1999 lahirlah teknologi *J2EE (Java 2 Enterprise Edition)* yang berbasis *J2SE* yang diawali dengan *Servlet* dan *EJB* kemudian diikuti *JSP*. *Java* juga menjadi lebih cepat populer di lingkungan *server side* dikarenakan kelebihannya di lingkungan *network* dan terdistribusi serta kemampuan *multi threading*. Sedangkan *J2ME (Java 2 Micro Edition)* dapat menghasilkan aplikasi

*mobile* baik *games* maupun *software* yang dapat dijalankan di peralatan *mobile* seperti ponsel (Somantri, 2004).

### 2.11.2 Karakteristik Java

*Sintaks Java* merupakan pengembangan dari bahasa *C/C++*.

Berikut adalah beberapa hal tentang pemrograman *Java* (Somantri, 2004):

1. Bersifat *portable* dan *platform independent*. Program *Java* yang telah ditulis akan dapat dieksekusi di mesin apapun dan sistem operasi apapun tanpa harus mengubah sedikitpun dari program tersebut.
2. Memiliki *garbage collection* yang dapat mendealokasikan memori secara otomatis.
3. Menghilangkan sifat pewarisan berganda yang terdapat pada *C++*.
4. Mengurangi *pointer aritmetika*. Pengaksesan lokasi memori secara langsung dengan menggunakan *pointer* memungkinkan program untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan. Untuk mengurangi kemungkinan kesalahan seperti ini penggunaan *pointer* pada *Java* telah dibatasi dengan menggunakan *reference*.
5. Memiliki *array* sejati.
6. Mengurangi kerancuan antara pemberian nilai pada *statement conditional*.

Selain itu *Java* memiliki karakteristik:

1. Berorientasi *Object*

2. *Robust*

3. *Portable*

4. *Multi Threading*

5. Dinamis

6. Sederhana

7. Terdistribusi

8. Aman

9. Netral secara arsitektur

10. *Interpreter*

11. Berkinerja tinggi

12. Free/bebas lisensi

### 2.11.3 Fitur-Fitur Java

Beberapa fitur yang ditawarkan *Java API* (Cahyono, 2006) antara lain sebagai berikut :

a) *Applet*

Program *Java* yang dapat berjalan di atas *browser*, yang dapat membuat halaman *HTML* lebih dinamis dan menarik.

b) *Java Networking*

Sekumpulan *API* (*Application Programming Interface*) yang menyediakan fungsi-fungsi untuk aplikasi jaringan, seperti penyediaan akses untuk *TCP*, *UDP*, *IP Address* dan *URL*. Tetapi *Java Networking*

tidak menyediakan akses untuk *ICMP* dikarenakan alasan *security* dan pada kondisi umum hanya administrator (*root*) yang bisa memanfaatkan protokol *ICMP*.

c) *Java Database Connectivity (JDBC)*

*JDBC* menyediakan sekumpulan *API* yang dapat digunakan untuk mengakses *Database* seperti *Oracle*, *MySQL*, *PostgreSQL*, *Microsoft SQL Server*.

d) *Java Security*

*Java Security* menyediakan sekumpulan *API* untuk mengatur *security* dari aplikasi *Java* baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates*.

e) *Java Swing*

*Java Swing* menyediakan sekumpulan *API* untuk membangun aplikasi *GUI (Graphical User Interface)* dan model *GUI* yang diinginkan bisa bermacam-macam, bisa model *Java*, model *Motif/CDE* atau model yang *dependent* terhadap *platform* yang digunakan.

f) *Java RMI*

*Java RMI* menyediakan sekumpulan *API* untuk membangun aplikasi-aplikasi *Java* yang mirip dengan model *RPC*

## 2.12 *MySQL*.

Menurut (Glass,2004) *MySQL* (dibaca **mai-es-ki-el** atau bisa juga **mai-se-kuel**) merupakan suatu *RDBMS* (*Relational Database*

*Management System), pada dasarnya digunakan untuk menyimpan berbagai macam informasi. Anda dapat hampir menyimpan apapun dalam Database ini. Informasi seperti alamat, nama, tanggal lahir, dan lain - lain. Seperti halnya Java yang menggunakan bahasa MySQL ini bersifat open source sehingga banyak yang menggunakan MySQL sebagai Database dibandingkan dengan produk komersial lainnya.*

Menurut (Welling, 2003) MySQL memiliki beberapa keuntungan diantaranya:

1. *Performance MySQL* sangat bagus dalam hal kecepatan menjalankan query
2. *Low Cost*; dikarenakan MySQL bersifat open source maka tidak memerlukan banyak biaya dalam penggunaanya.
3. *Easy Of Use*; MySQL dapat berjalan pada berbagai sistem operasi seperti halnya pada Microsoft Windows dan Linux
4. *Source Cose*; seperti halnya Java source code dalam MySQL dapat dimodifikasi sesuai kebutuhan.

### **2.12.1 Tipe Data dalam MySQL**

Menurut (Nugroho, 2005) Tipe-tipe data yang terdapat dalam MySQL antara lain:

1. Numeric

Tipe data ini mencakup bilangan bulat (INTEGER) dengan berbagai ukuran (INTEGER atau INT, dan SMALLINT) serta bilangan nyata

(REAL) dengan beberapa tingkat ketelitian (FLOAT, REAL, DOUBLE PRECISION).

## 2. Karakter – String

Tipe data ini mencakup tipe dengan panjang tetap yaitu CHAR(n) atau CHARACTER(n) dimana n adalah panjangnya karakter, serta tipe yang ukurannya berubah – ubah yaitu VARCHAR(n) atau HARVARYING(n).

## 3. Tanggal dan Waktu

Tipe data ini hanya dikenali di SQL-2. Tipe data DATE memiliki 10 posisi dan komponennya mencakup YEAR, MONTH, DAY dalam bentuk YYYY-MM-DD. Sedangkan TIME memiliki 8 posisi dengan komponen HOUR, MINUTE, SECOND dengan bentuk HH:MM:SS.

### 2.13 Database

Menurut (McLeod,2004) *Database* adalah suatu koleksi data komputer yang terintegrasi, diorganisasikan dan disimpan dengan suatu cara yang memudahkan pengambilan kembali. *Database* dapat dinyatakan sebagai suatu sistem yang memiliki karakteristik, antara lain :

- a. Merupakan suatu kumpulan *interrelated* data yang disimpan bersama tanpa mengganggu satu sama lain atau membentuk kerangkapan data.
- b. Kumpulan data dalam *Database* dapat digunakan oleh sebuah program aplikasi atau lebih secara optimal.
- c. Penambahan data baru, penghapusan data, modifikasi dan pengambilan kembali data dapat dilakukan dengan mudah dan terkontrol.

- d. Data merupakan suatu sumber yang sangat berguna bagi hampir di semua organisasi.

### 2.13.1 Komponen Database

a. *Entity*

Adalah orang, tempat, kejadian atau konsep yang informasinya disimpan atau direkam

b. *Attribute*

Setiap entity mempunyai *Attribute* atau sebutan untuk mewakili suatu entity. *Attribute* juga disebut data elemen, data field, data item.

c. *Data Value* (Nilai atau isi *Database* )

Data value adalah data actual atau informasi yang di simpan pada tiap data elemen atau *attribute*.

d. *Record/ Tuple*

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang *entity* secara lengkap. Satu record mewakili satu data atau informasi tentang seseorang, misalnya nomor karyawan, kota, dan tanggal masuk.

e. *File*

Kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, nama berbeda-beda data *valuensya*.

### 2.13.2 Kegunaan *Database*.

Penyusunan satu *Database* digunakan untuk mengatasi masalah-masalah pada penyusunan data (Cahyono, 2006), yaitu:

1. *Redudansi* dan inkonsistensi data
2. Kesulitan pengaksesan data
3. *Isolation* data dan standarisasi
4. *Multiple user* (banyak pemakai)
5. Masalah *security* (keamanan)
6. Masalah *Integration* (kesatuan)

Pada skripsi ini penulis menggunakan aplikasi *Database MySQL*, dikarenakan kemudahan dalam pemakaian, dan dapat berhubungan dengan berbagai aplikasi lainnya, termasuk aplikasi pemrograman.

## 2.14 Jaringan Komputer

### 2.14.1 Definisi Jaringan Komputer

Jaringan komputer adalah sekelompok komputer yang dihubungkan satu dengan yang lainnya dengan menggunakan protokol komunikasi tertentu melalui media transmisi tertentu sehingga dapat saling berbagi data, informasi, program-program, dan menggunakan bersama perangkat keras seperti pencetak (printer), cakram keras (harddisk), dan sebagainya.

Tujuan utama dibangunnya suatu jaringan komputer adalah untuk membawa data atau informasi dari sisi pengirim menuju

penerima secara cepat dan tepat tanpa adanya kesalahan melalui media transmisi tertentu (Purnomo, 2004).

#### 2.14.2 Arsitektur Jaringan Komputer

Ada tiga macam arsitektur jaringan komputer, yaitu *peer to peer*, *file server*, dan *client server*. Arsitektur ini menunjukkan bagaimana cara beberapa komputer dapat berkomunikasi satu dengan yang lainnya (Purnomo, 2004).

##### 1. *Peer to Peer*

Jaringan komputer dengan arsitektur *peer to peer* dibangun dengan cara menghubungkan setiap terminal (komputer, pencetak, dan perangkat keras lainnya) secara langsung sehingga masing-masing terminal dapat berbagi data dan perangkat lainnya dengan mudah. Pada arsitektur ini, setiap komputer memiliki peran dan derajat yang sama.

##### 2. *File Server*

Arsitektur *file server* adalah arsitektur jaringan komputer dengan satu atau lebih terminal khusus yang disebut *server* yang memiliki kapasitas harddisk yang sangat besar. Arsitektur ini memberikan keuntungan berupa efisiensi akses dan perawatan berkas data karena seluruh berkas yang sering digunakan bersama tersebut tersimpan pada satu tempat.

##### 3. *Client Server*

Arsitektur *file server* yang dikembangkan lebih lanjut sehingga server tidak hanya bertindak sebagai tempat penyimpanan data namun

juga sebagaimana pemrosesan data disebut dengan arsitektur *client server*. Prinsip kerjanya sangat sederhana, di mana *server* akan menunggu permintaan dari *client*, memprosesnya, dan memberikan hasilnya kepada *client*. Di lain pihak, *client* akan mengirimkan permintaan ke *server*, menunggu proses dan menerima hasilnya.

### 2.14.3 Definisi *Intranet*

*Intranet* adalah jaringan komputer untuk kalangan terbatas (biasanya suatu perusahaan) yang menggunakan teknologi *Internet* sehingga terbentuk lingkungan yang seperti *Internet* namun bersifat *privat* bagi kalangan yang bersangkutan.

*Privat* di sini berarti bahwa layanan-layanan *intranet* dalam sebuah perusahaan tertentu tidak dapat diakses oleh publik, namun hanya dapat diakses oleh orang-orang yang memiliki izin atau hak akses (biasanya dalam bentuk nama pengguna atau username dan sandi lewat atau *password*) yang biasanya terbatas pada karyawan dalam perusahaan tersebut (Purnomo, 2004).

## 2.15 Evaluasi Kelayakan Kredit/ Pembiayaan Konvensional yang lain.

Menurut (Hiemann,2002) sistem penilaian yang diperkenalkan adalah sistem penilaian menggunakan teknik skoring. Artinya, semua kriteria yang mempengaruhi penilaian kelayakan kredit diberi skor. Hasil skor keseluruhan mengindikasikan tingkat kelayakan suatu kredit diberikan.

Skor yang tinggi menggambarkan probabilitas yang tinggi dalam permasalahan peminjaman,

Salah satu model teknik skoring yang diperkenalkan menggunakan dua belas kriteria. Masing-masing kriteria telah diberi skor. Kriteria-kriteria tersebut antara lain: Penilaian berdasarkan biro pembiayaan, kapasitas kemampuan membayar kembali (% dari *disposable income*), jaminan, aset yang dapat dilikuidasi, sejarah pinjaman, kualifikasi professional, pengalaman usaha, bidang usaha yang dijalankan, umur, status perkawinan, tempat tinggal, dan jumlah tanggungan termasuk pasangan. Dibandingkan dengan penilaian secara manual, sistem penilaian dengan teknik skoring ini membantu mengurangi penunggakan 20 % sampai 40 %.

Di Amerika Serikat, *credit scoring* digunakan untuk penilaian pembiayaan konsumtif sudah puluhan tahun. Sampai beberapa tahun yang lalu, sistem analisa Kredit yang dipakai untuk perusahaan besar juga dipakai untuk perusahaan kecil. Terakhir ini sistem *credit scoring* dikembangkan untuk pembiayaan mikro (*s/d* \$100.000). Sebelumnya waktu yang dibutuhkan sampai pembiayaan dicairkan selama 12 jam atau lebih dengan biaya diantara US\$ 500 dan US\$1.800. Sekarang, jika pemohon mengajukan permohonan kredit lewat internet, waktu yang diperlukan untuk pemutusan pembiayaan dapat diperpendek hingga 15 menit. Rata-rata biaya pengelolaan permohonan pembiayaan turun dan diperkirakan sekitar US\$ 100 saja.

*Credit scoring* berdasarkan data statistik dan pengalaman terhadap paling sedikit masing-masing 500 kredit baik dan buruk untuk mempelajari

apa yang membedakannya. Teknik ini terutama cocok untuk bank yang nasabahnya mirip dari segi besarnya dan jenis usaha. *Credit scoring* tergantung juga dari informasi tentang kredit sebelumnya (termasuk kredit mikro dan penggunaan kartu kredit dsb.). Maka sangat cocok jika *credit scoring* dapat dihubungkan dengan *credit bureau*.

Saat ini ada beberapa bank umum yang sedang melakukan uji coba penggunaan metode skoring dalam melakukan penilaian kelayakan pemberian pembiayaan kepada UKM. Sebagai ilustrasi, Bank BNI di Jakarta mencoba menerapkan teknik skoring dan memperoleh beberapa manfaat, antara lain :

1. Waktu pemutusan pembiayaan menjadi lebih cepat
2. Pegawai yang sebelumnya dialokasikan untuk melakukan penilaian kelayakan UKM di kantor dapat dialokasikan menjadi tenaga pemasaran pembiayaan.
3. Portfolio pembiayaan kantor cabang di Jakarta meningkat, khususnya untuk portfolio kepada pasar UKM.

## 2.16 Penelitian Sejenis

Dalam membuat skripsi pengembangan sistem informasi persediaan barang ini, penulis juga telah melakukan pengkajian terhadap beberapa penelitian sejenis digunakan referensi yang ada pada perpustakaan Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, terdapat beberapa kekurangan dan kelebihan dari sistem informasi tersebut, seperti pada tabel berikut ini.

1. Analisa Dan Perancangan Sistem Informasi Kredit Mikro Di PT.BPRS MULYA ARTA, (Rahman,2007)

Penelitian tersebut memfokuskan pada kegiatan menganalisa sistem yang sedang berjalan, merancang *Database* dan membuat program sederhana yang hanya bertugas utnuk memasukkan / menginput data yang menjadi calon debitur dari BPRS Mulya Arta. Pada implementasinya menggunakan *Software Microsoft Access 2003* sebagai *Database* dan aplikasinya. Pemecahan masalah pada studi literatur ini adalah dengan pembuatan sistem informasi dengan berbasis komputer atau *Computized Based Information System (CBIS)*.

Tabel 2.4 Hasil Penelitian Sejenis

No	Kekurangan	No	Kelebihan
1.	Sistem yang dirancang masih belum efektif karena belum bisa diakses oleh user yang banyak.	1.	Sistem kredit ini sudah <i>Computize Based information System (CBIS)</i> .
2.	Belum maksimalnya penggunaan <i>Database</i> karena masih menggunakan <i>Ms.Access 2003</i> dan dapat diperkirakan bahwa akan banyak <i>bugs</i> dan data yang di kelola tidak banyak.	2.	Form laporan dalam aplikasi yang tersusun sistematis sehingga akan sangat memudahkan user dalam mengelola dan mempelajarinya.
3.	Belum adanya penghitungan mengenai layak tidaknya pemberian usaha nasabah.	3	Desain dan pengembangan aplikasi yang <i>user friendly</i> sehingga memudahkan dalam mengelola data dan informasi yang kompleks.

2. Sistem Pendukung Keputusan (*Decision Support System*) Penjurusan Program Studi Pada Sekolah Menengah Atas (Studi Kasus SMAN 1 Ciputat) (Masruroh, 2008)

Pembuatan yang dilakukan dengan menggunakan bahsa pemrograman *Borland Delphi 7.0* dan Database *MySQL*. Metodologi pengembangan sistem dilakukan dengan metode *SDLC* dan Metodologi model yang digunakan perhitungan *AHP*. Analisa penelitian yang dilakukan pada SMAN 1 Ciputat, yang dimana merupakan suatu lembaga pendidikan dibawah naungan DEPDIKBUD.

Tabel 2.5 Hasil Penelitian Sejenis

No	Kekurangan	No	Kelebihan
1.	Sistem yang dirancang masih belum bisa memenuhi kebutuhan institusi sekolah karena belum dirancang untuk tidak banyak user.	1.	Sistem ini sudah mencangkup semua standarisasi suatu perusahaan dari kriteria baku penjurusan sekolah.
2.	Belum lengkapnya fasilitas yang ada, seperti belum adanya klasifikasi kode barang standar berdasarkan pengkodean Barang, belum mencakup lokasi, pendistribusian, dan kondisi barang keluar, yang datanya sangat penting.	2.	Database yang digunakan adalah <i>MySQL</i> untuk laporannya, sangat mendukung untuk mengelola data dalam jumlah banyak dan dapat di integrasi dalam berbagai bahasa pemrograman.

3. Bentuk dari laporan yang kurang efektif, karena data-data yang dicantumkan kurang lengkap
- 3 Desain dan pengembangan aplikasi yang *user friendly* sehingga memudahkan dalam mengelola data dan informasi yang kompleks.

3. Rancang Bangun Sistem Pakar Untuk Mendeteksi Gizi Buruk Pada Balita (Aziz, 2010)

Penelitian tersebut dilakukan di Posyandu dan Puskesmas menggunakan Kartu Menuju Sehat (KMS) di Posyandu. Penelitian memfokuskan pada bagaimana cara kerja sistem pakar memproses sebuah *domain* permasalahan sehingga tercapai sebuah kesimpulan atau jawaban atas gizi buruk pada balita, dengan menggunakan *inference engine* dan teknik penelusuran *depth first search*. metodologi pengembangan sistem melalui pendekatan metodologi berorientasi *object* (*Object Oriented Modelling*) serta menggunakan tools pengembangan sistem *Unified Modelling Language (UML)* yang dikembangkan menggunakan model *Expert System Development Life Cycle* dengan menggunakan bahasa pemrograman *Java* dan *Database MySQL*. Pemecahan masalah pada studi literatur ini adalah dengan pembuatan aplikasi sistem pakar agar para ibu yang mempunyai anak balita dapat mendeteksi terjadinya gejala gizi buruk pada balita sejak dini dengan mengacu kepada Kartu Menuju Sehat (KMS) sehingga langkah pencegahan segera dilakukan.

Jadi pada studi literatur yang pertama sudah menerapkan sistem

komputerisasi dengan proses pemasukan data yang dapat tersimpan dengan baik, pada studi literatur yang kedua memiliki fokus pengembangannya dengan *SDLC* dan perhitungan pengujianya dengan perhitungan *AHP*. Sedangkan pada Studi literatur ketiga juga memiliki fokus ke perhitungan dan pertimbangan dengan Sistem pakar yang membatasi usia dalam kriteria yang di buat dan menggunakan metode pengembangan *UML* yang menggunakan bahasa pemrograman *Java*.

Tabel 2.6 Hasil Penelitian Sejenis

No	Kekurangan	No	Kelebihan
1.	Sistem yang dirancang masih belum bisa memenuhi kebutuhan user karena belum dirancang untuk kebutuhan banyak usia anak kecil.	1.	<i>Database</i> yang digunakan adalah <i>MySQL</i> untuk laporannya, sangat mendukung untuk mengelola data dalam jumlah banyak.
2.	Belum lengkapnya fasilitas yang ada, seperti belum adanya fasilitas link untuk jaringan dan print untuk mencetak hasil perhitungan maupun rekomendasi yang dihasilkan yang datanya sangat penting.	2.	<i>JAVA</i> memiliki kemampuan yaitu disupport oleh banyak <i>Database</i> sehingga membuat halaman aplikasi yang menggunakan data dari <i>Database</i> dapat dengan sangat mudah dilakukan, dan sangat membantu dalam mengolah data-data pembiayaan Usaha .
3.	Bentuk dari laporan yang kurang efektif, karena data-data yang dicantumkan kurang lengkap.	3	<i>Design</i> dan pengembangan aplikasi yang <i>user friendly</i> sehingga memudahkan dalam mengelola data dan informasi yang kompleks.

Maka penulis mencoba untuk membuat suatu pengujian kelayakan usaha untuk pembiayaan yang akan diberikan atau direalisasikan dengan logika *Fuzzy* yang dapat memberikan masukan dan keputusan kelayakan dari aspek penilaian meliputi kinerja nasabah dan usaha sesuai dengan kriteria standar perusahaan dilihat dari hasil survey dilapangan yang kemudian di *input* kedalam sistem.



## BAB III

### METODOLOGI PENELITIAN

Seperti yang telah dibahas pada bab 1, dalam sistem pendukung keputusan kelayakan pembiayaan UKM dengan logika *Fuzzy* ini, ada beberapa tahapan yang dilakukan, diantaranya.

#### 3.1 Pengumpulan Data

Dalam penelitian ini penulis menggunakan tiga metode untuk mengumpulkan data, yaitu observasi, wawancara dan studi pustaka.

##### 3.1.1 Observasi

Observasi merupakan teknik pengumpulan data yang efektif untuk mempelajari sebuah sistem. Dalam observasi, dilakukan teknik penemuan fakta dimana analisis sistem turut berpartisipasi atau menyaksikan seseorang yang sedang melakukan aktivitas untuk mempelajari sistem (Whitten, 2004).

Observasi dilakukan dengan mengumpulkan data dan informasi dengan cara mengamati langsung penilaian kelayakan pembiayaan yang dilakukan BPRS Wakalumi dalam memberikan pembiayaan kepada usaha kecil menengah pada saat magang. Observasi dilaksanakan di BPRS Wakalumi, Komp. Ciputat Mutiara Center Blok B1 Jl. Dewi Sartika – Ciputat 15411

### a. Gambaran Umum BPRS Wakalumi

Bank Syariah Wakalumi didirikan oleh Yayasan Wakalumi (Wakaf Karyawan dan Alumni Muslim Citibank), yang bergerak dalam bidang sosial dan pendidikan bagi anak yatim, miskin dan dhuafa khususnya. Tujuan didirikannya perseroan selain sebagai unit bisnis yang professional dan islami, juga untuk menyediakan dana berkesinambungan guna mendukung kegiatan yayasan. Perseroan resmi beroperasi secara konvensional sebagai BPR sejak 9 April 1990.

Berdasarkan Akte No. 78 Notaris B.R.A.Y Mahyastoeti Notonagoro, SH tanggal 9 Juni 1994, perseroan menyetujui masuknya Bank Muamalat Indonesia sebagai pemegang saham dengan kepemilikan sebesar 49 persen dan menyetujui perubahan system operasional menjadi syariah. Dengan masuk serta adanya bantuan teknis dan manajemen dari Bank Muamalat Indonesia, kinerja Bank Syariah Wakalumi semakin baik.

Setelah melalui transisi untuk melakukan konversi system operasional, maka sejak tahun 1995, perseroan resmi beroperasi dengan system syariah. Selanjutnya mulai tahun 2003, keterkaitan antara Bank Syariah Wakalumi dengan Bank Muamalat Indonesia ditingkatkan dengan menjadikan Bank Syariah Wakalumi sebagai mitra kerja Bank Muamalat Indonesia. Kini Bank Syariah Wakalumi memiliki 5 kantor kas yang tersebar di wilayah kabupaten Tangerang.

Bidang Usaha; Bank Pembiayaan Rakyat Syariah adalah Bank Desa atau Bank Pasar yang berlandaskan syariah. BPRS tidak melayani lintas Pembayaran. Adapun pandangan manajemen perusahaan menjadi mitra yang dipercaya dan bersahabat, memasyarakatkan fungsi bank dalam membantu pengembangan usaha kecil dan menengah yang turut serta membantu dalam program pemerintah untuk mengembangkan pengusaha kecil dan menengah.

Sejak berdiri perseroan telah mengalami beberapa kali perubahan modal dasar, berawal dari Rp 300 Juta menjadi Rp. 1,5 Miliar. Kemudian pada tahun 2000 meningkat menjadi Rp. 3 Miliar. Kepemilikan perseroan terbuka bagi keluarga besar yayasan wakalumi , bank Wakalumi, Bank Muamalat maupun mereka yang memiliki komitmen yang sama untuk mengembangkan ekonomi umat.

Saat ini wakalumi yang memiliki semboyan “Bersama Melayani Umat” dalam pemenuhan kebutuhan modal maupun investasi usaha, baik itu masyarakat maupun para pengusaha dengan melakukan pembukaan kantor kas di beberapa desa wilayah Tangerang yaitu :

1. Kantor Pusat : Komplek Ciputat Mutiara Center. Blok B1, Jl. Dewi Sartika – Ciputat. Telp. (021) 740-1667, 749-0874 Fax. (021) 744-2788
2. Kantor Kas Serpong : Jl. raya Serpong No 54 B. Serpong Telp/Fak. (021) 75870401.

3. Kantor Kas Cikupa : Jl Raya Serang Km 14,8 Cikupa  
Telp/faks. (021) 59402224.

4. Kantor Kas Balaraja : Jl Raya Serang Km 23 Balaraja  
Telp/Faks (021)

5. Kantor kas Podok Aren : Jl Raya Bintaro Cager No. 121 Pd  
Aren Telp/faks. (021) 73882465.

6. Kantor Kas Cisoka : Jl. Raya Cikupa 153 Cisoka  
Telp/Fax.(021) 5975-0499

#### **b. Visi, misi dan Motto BPRS Wakalumi**

Adapun visi, misi dan motto BPRS Wakalumi adalah

##### 1. Visi

Visi BPRS Wakalumi ialah menjadikan BPR Syariah  
terbaik dan terpercaya

##### 2. Misi

a. Memberdayakan ekonomi umat dengan fokus usaha mikro,  
kecil dan menengah.

b. Memberikan pelayanan unggul dan amanah bagi para mitra  
usaha.

c. Memiliki sistem dan cara kerja yang unggul dangan sumber  
daya insani yang profesional, kompeten, handal dan  
menjunjung tinggi *Ukhuwah Islamiyah*.

d. Memberikan manfaat optimal bagi para *Stakeholder*.

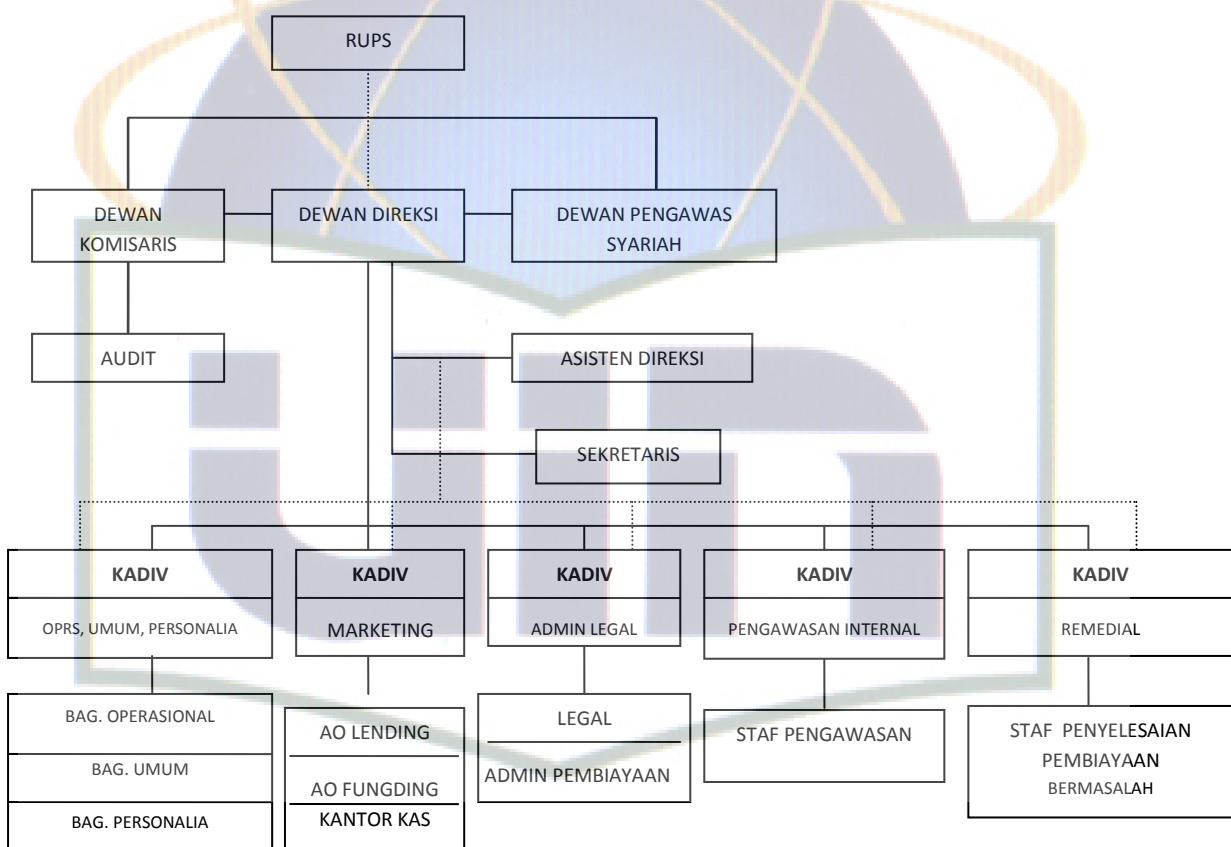
e. Memberikan kontribusi nyata bagi negara dan bangsa

### 3. Motto

Adapun motto BPR Syariah Wakalumi adalah membangun

kualitas hidup yang hasanah

### c. Struktur Organisasi



Gambar 3.1 Struktur Organisasi BPRS Wakalumi

(Sumber: Keterangan Kadiv Marketing BPRS Wakalumi, Tahun 2010)

### 3.1.2 *Interview / Wawancara*

Wawancara atau *interview* merupakan teknik penelusuran fakta dimana analis sistem mengumpulkan informasi dari individu - individu melalui interaksi *face to face* (Whitten, 2004).

Pada tahap ini, dilakukan wawancara dengan analisis pembiayaan yang disebut komite dari bagian Marketing PT BPRS WAKALUMI dan survey yang ditugaskan untuk mempertimbangkan dan menilai kebutuhan-kebutuhan bagi pemohon pembiayaan.

Alat yang digunakan untuk wawancara berupa alat tulis, dan alat perekam suara yaitu *handphone* dengan kapasita memori 2Gb.

### 3.1.3 Studi Pustaka

Pengumpulan data biasanya diawali dengan mengumpulkan informasi yang berhubungan dengan masalah penelitian. Informasi-informasi tersebut dapat diperoleh melalui peninjauan *literature* yang relevan (Gulo, 2002).

Studi literatur dalam penelitian ini dilakukan dengan cara mempelajari buku karangan dari para manajemen Bank Syariah diantaranya Adi Warman Karim, Muhammad, dan sumber-sumber lain untuk menunjang dalam pembangunan sistem pakar ini. Adapun daftar buku-buku dan situs-situs web yang menjadi referensi dalam penelitian ini dapat dilihat pada daftar pustaka.

### 3.2 Metode Pengembangan Sistem

Pengembangan sistem dapat diartikan sebagai sebuah proses pengembangan terstandarisasi yang mendefinisikan satu set aktifitas, metode, praktik terbaik, dan perangkat terotomatisasi yang akan digunakan oleh para pengembang sistem dan manajer proyek untuk mengembangkan dan berkesinambungan memperbaiki sistem informasi dan perangkat lunak (Whitten, 2004).

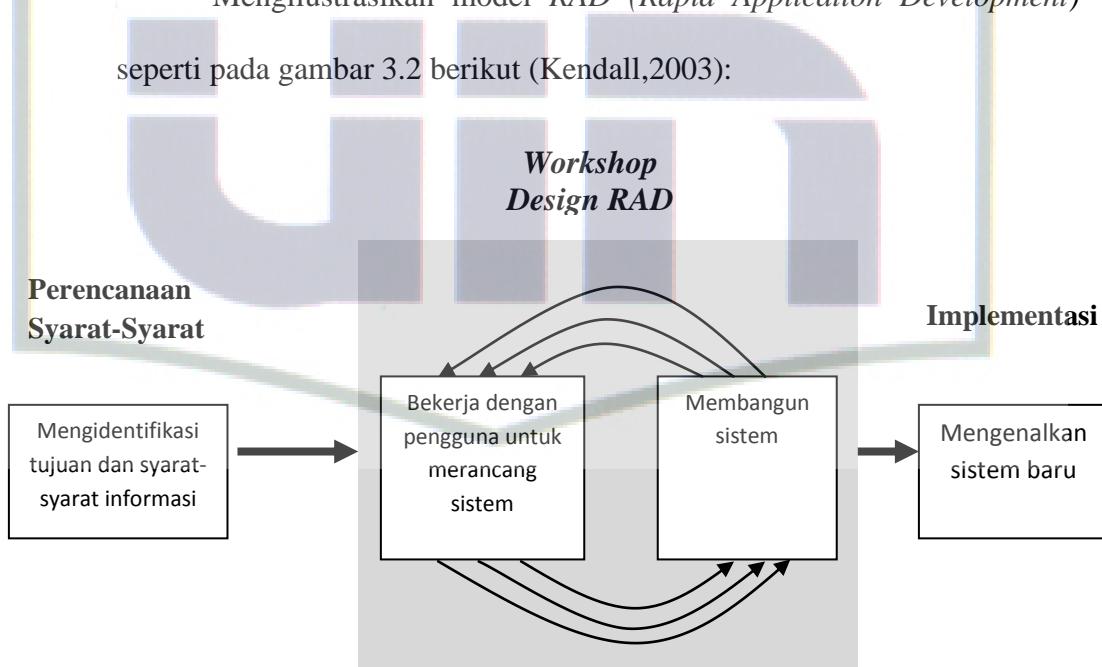
Penulis Mengembangkan *Prototyping* sistem penilaian kelayakan pembiayaan dengan menggunakan logika *Fuzzy*

*Prototype* merupakan teknik pengumpulan informasi dimana akan mempercepat reaksi dari penggunaan dan manajemen serta saran-saran penggunaan mengenai perubahan atau pemecahan masalah sehingga dapat dikembangkan lebih lanjut untuk menghasilkan sistem yang mendetail. *Prototype* menekankan pada fungsi kerja sistem secara keseluruhan. Sedangkan *RAD* merupakan model proses pengembangan perangkat lunak yang menekankan siklus pengembangan yang sangat pendek. Oleh karena itu metode pengembangan yang digunakan untuk *Prototype* ini adalah *Rapid Application Development (RAD)*. *RAD* dan *Prototyping* sama-sama mempersingkat waktu dalam perancangan dan sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah sangat cepat (Kendall and Kendall, 2003).

Dibandingkan dengan metode lainnya, implementasi fase *RAD* dalam beberapa cara tidak terlalu ditekankan, karena penggunaan terbantu

untuk merancang aspek-aspek sistem perusahaan dan sangat menyadari perubahan yang harus dilakukan. Sedangkan metode lainnya misalnya *SDLC (Sistem Development Life Cycle)* membutuhkan waktu yang sangat panjang selama pengembangan dan perancangan jika menganalisis terpisah jauh dari pengguna, sehingga memungkinkan produk finalnya berubah dan berbeda dengan apa yang diantisipasi selama beberapa bulan. Selain itu *RAD* adaptasi dari metode pengembangan *Sequensial Linear* yang menekankan siklus perkembangan yang sangat pendek, dimana perkembangan cepat dapat dicapai dengan menggunakan pendekatan konstruksi berbasiskan komponen (Pressman, 2002).

Mengilustrasikan model *RAD* (*Rapid Application Development*) seperti pada gambar 3.2 berikut (Kendall,2003):



Gambar 3.2 Fase-fase *RAD*

### 3.2.1 Fase Perencanaan Syarat-Syarat

Dalam fase ini, pengguna dan menganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Fase ini memerlukan peran aktif dari kedua kelompok tersebut. Selain itu juga melibatkan penggunaan dari beberapa level yang berbeda dalam organisasi. Orientasi dalam fase ini ialah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan perusahaan.

### 3.2.2 Fase Workshop Design

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Selama *workshop design RAD*, pengguna merespon *working Prototype* yang ada dan *analyst* memperbaiki modul-modul yang dirancang menggunakan perangkat lunak berdasarkan respon pengguna.

Adapun metode *design Prototype* sistem *Fuzzy* yang digunakan terdiri dari beberapa tahapan (Kusumadewi, 2004):

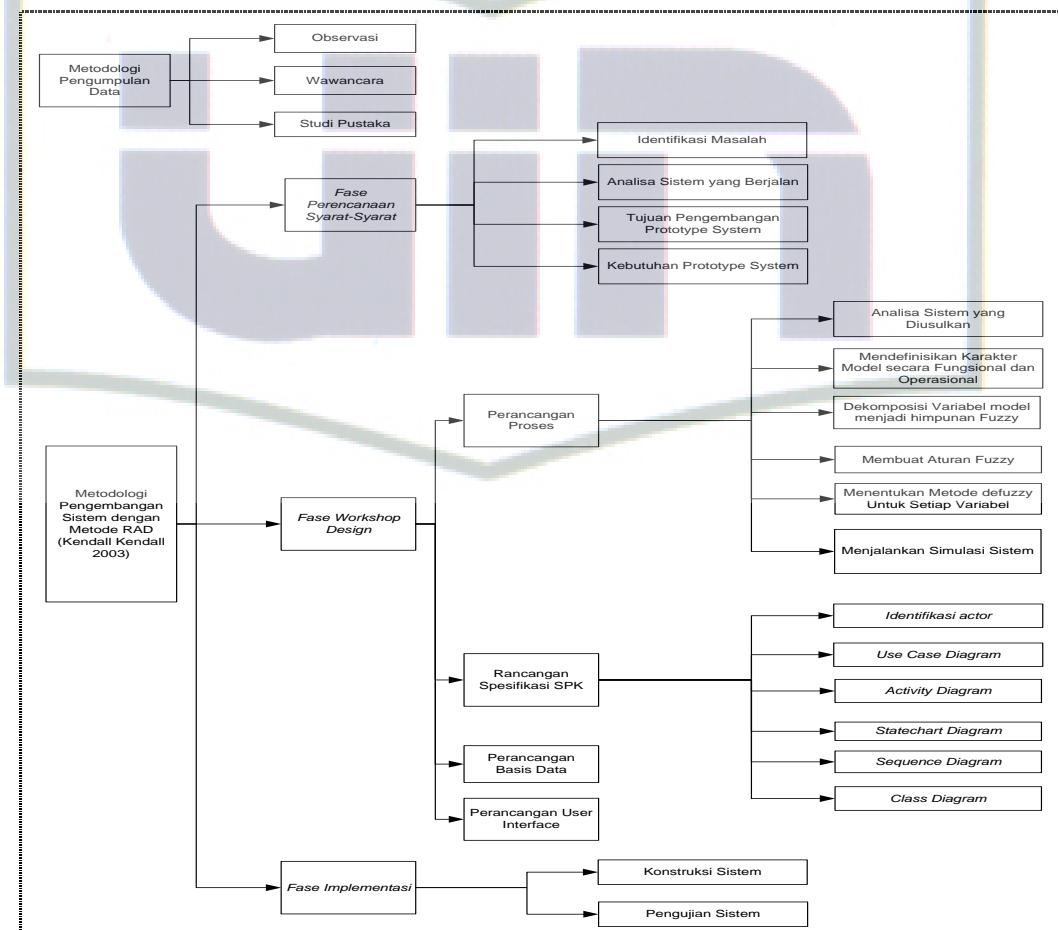
1. Mendefinisikan karakteristik model secara fungsional dan operasional.
2. Melakukan dekomposisi variabel model menjadi himpunan *Fuzzy*.
3. Membuat aturan *Fuzzy*.
4. Menentukan metode *defuzzy* untuk tiap-tiap variabel solusi.

5. Menjalankan simulasi sistem.
6. Pengujian, pengaturan dan validasi model.

### 3.2.3 Fase Implementasi

Pada tahap ini dilakukan pengujian terhadap sistem dan melakukan pengenalan terhadap sistem. Dalam hal ini sistem penunjang keputusan kelayakan pembiayaan ini di uji dan dikenalkan pihak BPRS WAKALUMI.

### 3.3 Kerangka Berfikir



Gambar 3.3 Kerangka Berfikir (*Logical Frame Work*)

## BAB IV

### ANALISA DAN PEMBAHASAN

Sebagaimana telah diuraikan dalam bab sebelumnya, metodologi yang digunakan untuk mengembangkan sistem penunjang keputusan kelayakan pembiayaan usaha kecil menengah (UKM) pada BPRS WAKALUMI adalah metode *Rapid Application Development (RAD)*. Bab ini akan membahas lebih rinci implementasi metodologi *RAD* dalam membangun sistem.

Sebelum menguraikan implementasi *RAD*, terlebih dahulu akan diuraikan analisa sistem yang berjalan dan analisa kebutuhan sistem yang berjalan.

#### 4.1 Fase Perencanaan Syarat-Syarat

Sebagaimana telah diuraikan pada bab sebelumnya, fase ini adalah fase identifikasi tujuan, kebutuhan serta berorientasi untuk menyelesaikan permasalahan-permasalahan yang ada.

##### 4.1.1 Identifikasi Masalah

Tahap mendefinisikan masalah adalah tahap untuk menentukan masalah apa yang harus diselesaikan dengan menggunakan sistem aplikasi yang akan dibuat. Dari hasil analisa diketahui bahwa pembiayaan dapat diberikan kepada Calon Debitur/pemohon selama menurut penilaian bank

yang dilakukan oleh staff komite, layak untuk menerima fasilitas pembiayaan.

Permasalahan yang belum dapat ditangani sepenuhnya oleh BPRS adalah memberikan keputusan penilaian yang lebih akurat dengan mengakomodasi aturan-aturan (*rule*) bisnis yang ada secara fleksibel.

Pada penelitian ini, penulis menitik beratkan pada proses penilaian hasil data *survey*, bukan penilaian petugas pada *survey*.

Syarat-syarat pengajuan pembiayaan usaha:

Dokumentasi yang diperlukan dalam proses pembiayaan

1. Formulir permohonan pembiayaan.
2. Kelengkapan Umum.
  - a. Pembiayaan yang sifatnya kolektif dan pengusaha kecil yang belum berbadan usaha wajib melengkapi :
    1. Fotocopy KTP suami/istri
    2. Fotocopy Kartu Keluarga dan Surat Nikah
  3. Surat keterangan belum menikah (bagi yang belum)
  4. Surat keterangan domisili apabila pemohon bertempat tinggal tidak menetap.
  5. Peta lokasi rumah
  6. Daftar barang yang akan dibeli apabila pembiayaan yang dimaksudkan untuk pembelian suatu barang.
- b. Pengusaha kecil yang belum berbadan usaha perlu menambahkan kelengkapan lain berupa:

1. Laporan keuangan sederhana (dapat dibuatkan oleh *Account Officer*).
  2. Fotocopy Surat Perintah Kerja (SPK) apabila tujuan pembiayaan adalah untuk pemenuhan modal pelaksanaan suatu proyek.
  3. Fotocopy Surat Izin Usaha Perdagangan (SIUP), Nomor Pokok Wajib Pajak (NPWP), dan Tanda Daftar Perusahaan (TDP).
  4. Surat Keterangan domisili usaha.
- c. Perusahaan yang telah berbadan usaha wajib melengkapi dokumen berupa :
1. Fotocopy KTP pemohon dan pengurus lain yang memberikan persetujuan sesuai akte pendirian maupun perubahannya.
  2. Proposal yang berisi sejarah ringkas perusahaan, data pemasok dan pelanggan, struktur organisasi, serta perincian asset yang dimiliki oleh perusahaan.
  3. Laporan Keuangan (Neraca dan rugi/laba) dua tahun terakhir dan atau dua bulan terakhir.
  4. Fotocopy Akte Pendirian dan perubahannya.
  5. Fotocopy surat pengesahan anggaran dasar dari Menteri Kehakiman (Perusahaan) atau Menteri Koperasi dan PPK (Koperasi).
  6. Fotocopy Surat Izin Usaha Perdagangan (SIUP), Nomor Pokok Wajib Pajak (NPWP), dan Tanda Daftar Perusahaan (TDP).

7. Fotocopy Tanda Daftar Rekanan (TDR) untuk perusahaan yang bergerak di bidang jasa konstruksi.
8. Surat keterangan domisili usaha
9. Fotocopy Surat Perintah Kerja (SPK).
10. Daftar barang yang akan dibeli.
3. Bukti-bukti kepemilikan agunan dan atau keterangan jaminan lainnya.

#### 4.1.1.1 Identifikasi Lingkup Sistem

Pada tahap ini dijelaskan kriteria-kriteria yang dijadikan acuan dalam proses penilaian kelayakan pembiayaan UKM. Kriteria-kriterianya adalah :

a. Jaminan

Tabel 4.1 Kriteria Syarat Jaminan

Benda tak bergerak	Benda Bergerak	Benda tak Bergerak
1. Tanah 2. Tanah beserta bangunan 3. Satuan rumah susun Syarat – syaratnya : <ul style="list-style-type: none"> <li>• Tanah dan atau beserta bangunan diatasnya yang dapat dijaminkan adalah : tanah yang berdasarkan Sertifikat Hak Milik (SHM), Sertifikat Hak Guna Bangunan (SHB), Sertifikat Hak Pakai (SHP) dan Girik.</li> <li>• Milik keluarga inti</li> </ul>	1. Kendaraan bermotor 2. Emas 3. Surat berharga (Deposito, Tabungan dan Tagihan). Syarat – syaratnya : <ul style="list-style-type: none"> <li>• Milik keluarga inti.</li> </ul> Khusus untuk tagihan: <ul style="list-style-type: none"> <li>• Milik nasabah</li> <li>• Merupakan jaminan tambahan.</li> </ul>	1. Personal <i>Guarantee</i> 2. <i>Coorporate Guarantee</i> Syarat – syaratnya : <ul style="list-style-type: none"> <li>• Orangnya dikenal oleh pejabat Bank , merupakan jaminan penunjang.</li> </ul>

Seorang pemohon pembiayaan yang mengajukan pinjaman harus memiliki jaminan yang sesuai dengan nilai yang akan dibiayainya. Hal yang berkaitan dengan analisis ini

dapat dilihat dari segi Jaminan dan Yuridis yaitu apakah jaminan yang diberikan cukup baik dalam arti dapat dipasarkan sehingga dapat dijual dan menilai kelayakan calon nasabah beserta usahanya dari segi hukum.

b. Kemampuan

Kemampuan maksud disini dengan dilihat dari kemampuan pemohon dalam kehidupan sehari-hari, dilihat dari penghasilan dan pengeluaran perbulan, kepemilikan tempat tinggal, kondisi tempat tinggal, dan jumlah tanggungan.

c. Usaha

Usaha disini dilihat dari analisa keuangan usaha dengan melihat laporan keuangan (neraca dan rugi laba), analisa management dengan melihat manajerial pengelola dalam usahanya, analisa industri dengan membandingkan calon nasabah dengan usaha sejenis, analisa bisnis dengan melihat kondisi calon usaha nasabah yang dihubungkan dengan usaha lain yang langsung berhubungan seperti penyedia bahan baku, proses sampai dengan barang siap dipasarkan, dan analisa resiko makro dengan menganalisis kondisi *poleksosbudhankam*.

d. Karakter

Karakter dalam hal ini dilihat dari perilaku sehari-hari, penampilan, interaksi dalam bermasyarakat dan ibadahnya.

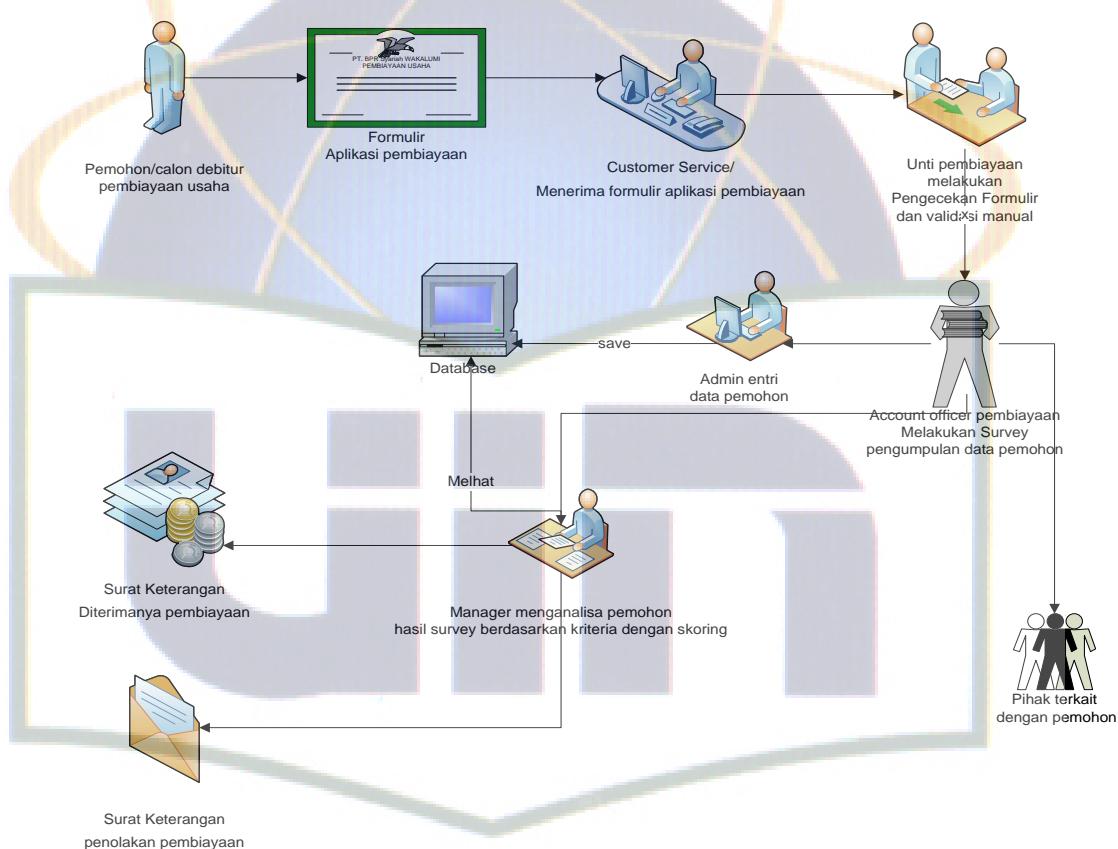
#### 4.1.2 Analisa Sistem Yang Berjalan

PT. BPRS WAKALUMI adalah perusahaan bergerak dibidang perbankan syariah yang mulai berkembang salah satu fasilitas yang ada di BPRS WAKALUMI adalah pemberian pembiayaan modal usaha. Sistem penilaian kelayakan pembiayaan usaha pada sistem yang berjalan di PT.BPRS Syariah sebagai berikut:

1. Prosedur memperoleh pembiayaan diawali dengan Calon Debitur menanyakan persyaratan kebagian layanan nasabah disebut *Customer Service (CS)* dan mengisi formulir aplikasi pembiayaan.
2. Calon Debitur mengisi formulir permohonan pembiayaan usaha, dan menyerahkan persyaratan. Unit pembiayaan akan melakukan pengecekan dan validasi data Jika dokumen tidak lengkap, maka Calon Debitur harus melengkapi persyaratan.
3. *Account Officer* akan melakukan silaturahim dan *survey* ke pemohon dan pihak terkait mencocokkan data yang telah didapat untuk di verifikasi kebenaran dan keabsahan dokumen syarat pengajuan pembiayaan
4. Setelah diverifikasi keabsahan dokumen-dokumen maka proses selanjutnya maka admin entri data identitas pemohon kedalam sistem
5. Setelah *Account Officer* melakukan keabsahan dokumen bersama manager melakukan penilai kelayakan pembiayaan Calon Debitur

tersebut, penilaian dilakukan dengan menganalisis seluruh data kinerja dan usaha Calon Debitur dengan cara manual.

6. Pengklasifikasian kelayakan jika Calon Debitur tersebut layak memperoleh pembiayaan dan memperoleh buku Debitur. Jika tidak, maka pihak bank memberikan informasi alasan penolakan.



Gambar 4.1 *Rich Picture* Analisa Sistem yang Berjalan

#### 4.1.3 Tujuan Pengembangan *Prototype* Sistem

Setelah dilakukan penelitian pada sistem yang sedang berjalan ditemukan beberapa kendala yang mungkin terjadi pada sistem penilaian kelayakan pembiayaan. Salah satunya adalah menggunakan teknik skoring. Perhitungan tersebut bersifat kaku (tidak fleksibel), sehingga

menyebabkan para pengusaha sulit untuk mencapainya karena keterbatasan yang dimilikinya, selain itu perhitungan yang ada masih bersifat manual.

Dengan melihat permasalahan tersebut, maka diperlukan sebuah sistem yang dapat mengakomodasikan aturan-aturan bisnis yang berlaku di BPRS WAKALUMI. Sistem diharapkan bersifat fleksibel dalam menentukan kelayakan pemberian pinjaman. Oleh karena itu, penulis memutuskan membangun sistem baru berbasis logika *fuzzy*.

Pengembangan *prototype* sistem penunjang keputusan kelayakan pemberian pinjaman UKM bertujuan untuk membantu peran analis atau pihak BPRS WAKALUMI dalam melakukan proses penilaian kelayakan dengan menggunakan logika *Fuzzy*.

#### 4.1.4 Kebutuhan *Prototype* Sistem

1. Memerlukan *input* yang mudah dilakukan.
2. Memerlukan sistem yang mampu memproses data-data yang di *input*, data masukan yang dibutuhkan adalah kondisi kriteria-kriteria yang meliputi data kinerja nasabah dan usahanya.
3. Memerlukan sistem yang mampu memberikan penilaian kelayakan pemberian pinjaman usaha kecil menengah.

## 4.2 Fase Workshop Design

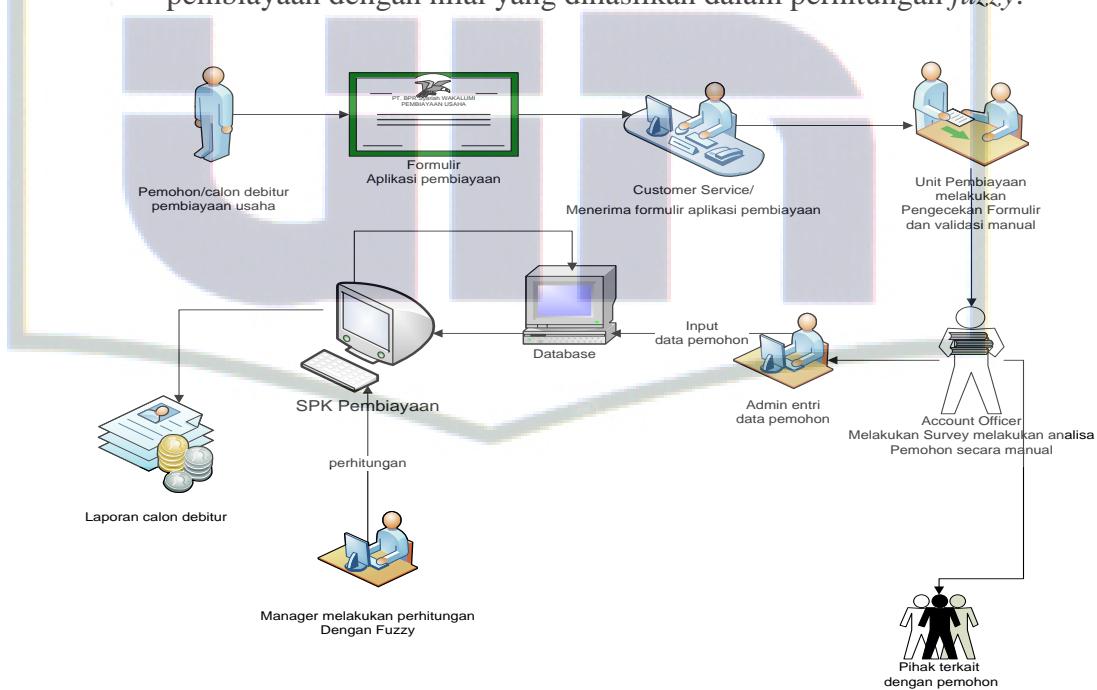
Dalam Fase ini, penulis secara aktif berinteraksi dengan pengguna untuk mengembangkan *prototype* sistem.

### 4.2.1 Sistem Yang Di Usulkan

Untuk memecahkan masalah keefektifan dalam menganalisa Calon Debitur pemberian Usaha, perlu adanya sistem yang dapat menganalisa langsung dalam menilai kelayakan pembiayaan untuk memudahkan Calon Debitur mendapatkan pinjaman. Oleh karena itu, diperlukan suatu sistem penunjang keputusan kelayakan pembiayaan yang lebih fleksibel dalam melakukan penilaian. Dengan menggunakan metode *Fuzzy*, ada beberapa kemungkinan nilai yang akan dihasilkan, sehingga lebih menguntungkan Calon Debitur. Berikut hal-hal yang perlu dirancang untuk pendukung sistem usulan :

1. Prosedur awal sampai pengecekan formulir dan validasi manual masih sama, Account manager melakukan analisa pemohon menerima dengan *survey* dengan mendapatkan data-data yang dibutuhkan, admin menginput data pemohon kedalam sistem Setelah proses penginputan selesai sistem akan menampilkan data hasil *input* tadi sebagai konfirmasi, apakah ada kesalahan dalam pengisian atau tidak, jika ada kesalahan maka *user* dapat melakukan proses *edit* pada sistem, jika tidak ada maka data secara otomatis akan tersimpan di dalam *Database*.

2. Manager tidak lagi melakukan analisa kelayakan pembiayaan secara manual dan, melainkan proses penilaian akan secara otomatis dihitung oleh sistem berdasarkan informasi kebenaran data pemohon. Dalam proses perhitungan dalam sistem penunjang keputusan pembiayaan, Manager hanya menyesuaikan dari data yang ada dengan kriteria yang sesuai dengan pemohon. Sehingga menghemat waktu dan memudahkan pemantauan dalam menganalisa kebenaran data untuk menilai dengan *Fuzzy* yang bersifat fleksibel dalam hasil nilai perhitungannya. *Output* sistem berupa hasil perhitungan penilaian masing-masing pemohon, laporan Calon penerima kelayakan pembiayaan dengan nilai yang dihasilkan dalam perhitungan *fuzzy*.



Gambar 4.2 Rich Picture Sistem yang Diusulkan

#### 4.2.2 Perancangan Proses Pada SPK Kelayakan UKM

Adapun perancangan proses disini merupakan analisis perhitungan untuk *design* sistem *Fuzzy* dengan metode Tsukamoto tahapan-tahapannya (Kusumadewi, 2002) adalah:

1. Mendefinisikan Karakteristik model secara fungsional dan operasional.
  2. Melakukan dekomposisi variabel model dengan.
  3. Menjadikan himpunan *Fuzzy*.
  4. Membuat aturan *Fuzzy*.
  5. Menentukan metode *defuzzy* untuk tiap-tiap variabel solusi
  6. Menjalankan simulasi sistem, pengujian-pengaturan dan validasi model.
- 4.2.2.1 Karakteristik Model Secara Fungsional dan Operasional**
- Tahap mendefinisikan Karakteristik model secara fungsional dan operasional adalah menentukan himpunan *Fuzzy* dan mendefinisikan beberapa *fuzzy* yang digunakan pada sistem ini.

Sistem kelayakan pembiayaan UKM ini dibangun oleh 5 Kriteria penilaian Variabel. Terdiri dari 4 Variabel *input* yang menjadi tolak ukur penilaian, yaitu variabel Jaminan, Usaha, Kemampuan dan Karakter serta 1 variabel Evaluasi penilaian. Semua kriteria digolongkan sebagai data *fuzzy*, karena data tidak bersifat mutlak melainkan sangat bersifat subjektif.

Ada tiga himpunan *fuzzy* yang digunakan oleh sistem ini untuk menilai setiap variabel *input*, yaitu Sangat Bagus (SB), Cukup Bagus (CB) dan Tidak Bagus (TB). Sedangkan untuk variabel *output* menggunakan lima himpunan *fuzzy*, yaitu Sangat Layak (SL), Layak Resiko Rendah (LRR) dan Sangat Tidak Layak (STL).

Klasifikasi skor himpunan *fuzzy* semua variabel tersebut terletak pada selang [0-10]. Klasifikasi variabel sistem ini dapat dilihat pada tabel 4.2 berikut:

Tabel 4.2: Variabel Penilaian Kelayakan Pembiayaan Beserta *Range*.

<b>Variabel</b>		<b>Fuzzy</b>	<b>Nilai</b>	<b>Range</b>
<b>Input</b>	1. Jaminan	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6-10 3-7 0-4
	2. Usaha	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6-10 3-7 0-4
	3. Kemampuan	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6-10 3-7 0-4
	4. Karakter	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6-10 3-7 0-4
<b>Output</b>	5. Evaluasi Penilaian	F	a. Sangat Layak (SL) b. Layak dengan Resiko Rendah (LRR) c. Layak dengan Resiko Tinggi (LRT) d. Kurang/ Tidak Layak (K/TL) e. Sangat Tidak Layak (STL)	9<X≤10 6<X≤9 4<X≤6 1<X≤4 0<X≤1

Sumber: data Sekunder

Pada tabel 4.2 nilai *linguistik* dari variabel *input* (Jaminan, Kemampuan, Usaha dan Karakter) menunjukkan maksimum, normal,

minimum. Pada keempat variabel *input* tersebut penulis menggunakan kata Sangat Bagus (SB) untuk menunjukkan nilai maksimum, Cukup Bagus (CB) untuk menunjukkan nilai normal, dan Tidak Bagus (TB) untuk menunjukkan nilai minimum.

Sedangkan nilai linguistik dari variabel *output* (evaluasi penilaian) penulis memakai kata Sangat Layak (SL) untuk nilai maksimum, Layak dengan Resiko Rendah (LRR), Layak Resiko Tinggi (LRT) untuk nilai normal, dan Kurang/Tidak Layak (K/TL), Sangat Tidak Layak (STL) untuk nilai minimum. Kata-kata tersebut hanya untuk alasan kepraktisan saja, sehingga memudahkan dalam menyebut nilai maksimum, normal, atau minimum suatu variabel.

Berdasarkan pengamatan yang dilakukan, petugas dapat memberikan penilaian terhadap kriteria-kriteria penilaian. Untuk mempermudah penilaian, petugas juga melihat beberapa aspek yang bagus untuk setiap variabel. Aspek-aspek tersebut penulis menyebutnya parameter. Parameter setiap variabel berbeda-beda. Untuk mengetahui besarnya pengaruh masing-masing parameter dalam setiap variabel, perlu dilakukan pembobotan untuk setiap variabel. Pembobotan dilakukan dan sudah ditentukan oleh BPRS Wakalumi. Klasifikasi parameter, bobot, serta indikator setiap parameter dapat dilihat pada Lampiran 1. Sistem ini menggunakan fungsi keanggotaan linear dan segitiga untuk memperoleh derajat keanggotaan suatu nilai *input* dan *output* dalam himpunan *fuzzy*. Pendekatan dengan representasi segitiga dan linear. Representasi linear

diambil dengan alasan nilai batas bawah dan atas suatu himpunan *fuzzy* pada variabel sudah baku berdasarkan ketentuan BPRS. Ada 2 keadaaan himpunan *fuzzy* yang linear, yaitu Tidak Bagus (TB) dan Sangat Bagus (SB). Pertama, penulis menggunakan linear turun untuk TB. Garis lurus dimulai dari domain dengan derajat keanggotaan tertinggi [1] pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah [0]. Kedua, penulis menggunakan linear naik untuk SB. Garis lurus dimulai dari domain dengan derajat keanggotaan terendah [0] pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih tinggi [1].

Adapun penulis menggunakan representasi segitiga karena ada 1 keadaan himpunan *fuzzy* yang nilai tengahnya harus maksimal (memiliki derajat keanggotaan 1), dan jika bergerak turun derajat keanggotaannya kurang dari satu. yaitu himpunan *fuzzy* Cukup Bagus (CB). Sehingga pendekatan segitiga dan linear adalah pendekatan yang paling cocok pada kasus ini.

#### 4.2.2.2 Melakukan Dekomposisi Variabel Model Menjadi Himpunan *Fuzzy*

Ada 5 variabel *fuzzy* yang akan dimodelkan, yang dibagi menjadi dua (2) bagian yaitu variabel *input* dan variabel *output*.

a. Variabel *Input*

Variabel *input* terdiri dari Jaminan, Usaha, Kemampuan dan Karakter.

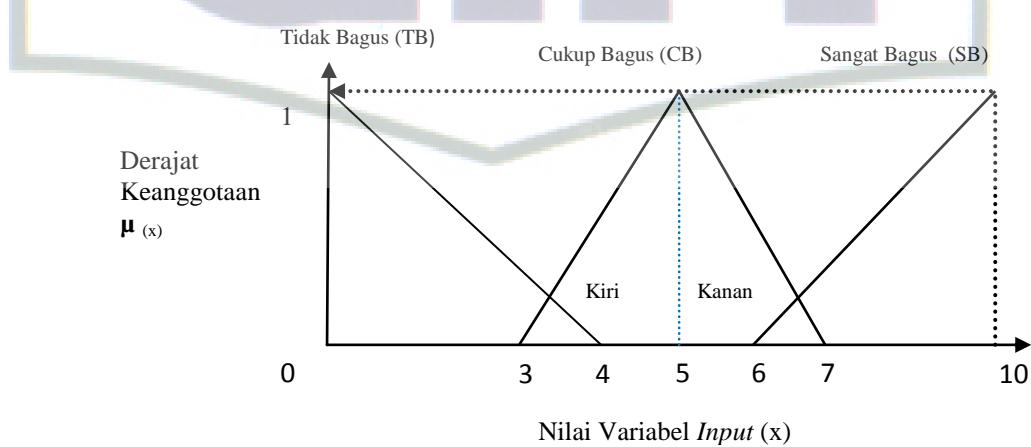
Keempat variabel tersebut mempunyai tiga himpunan *fuzzy* (nilai dan range) yang sama. Lihat Tabel 4.3

Tabel 4.3 Variabel *Input* Beserta Range

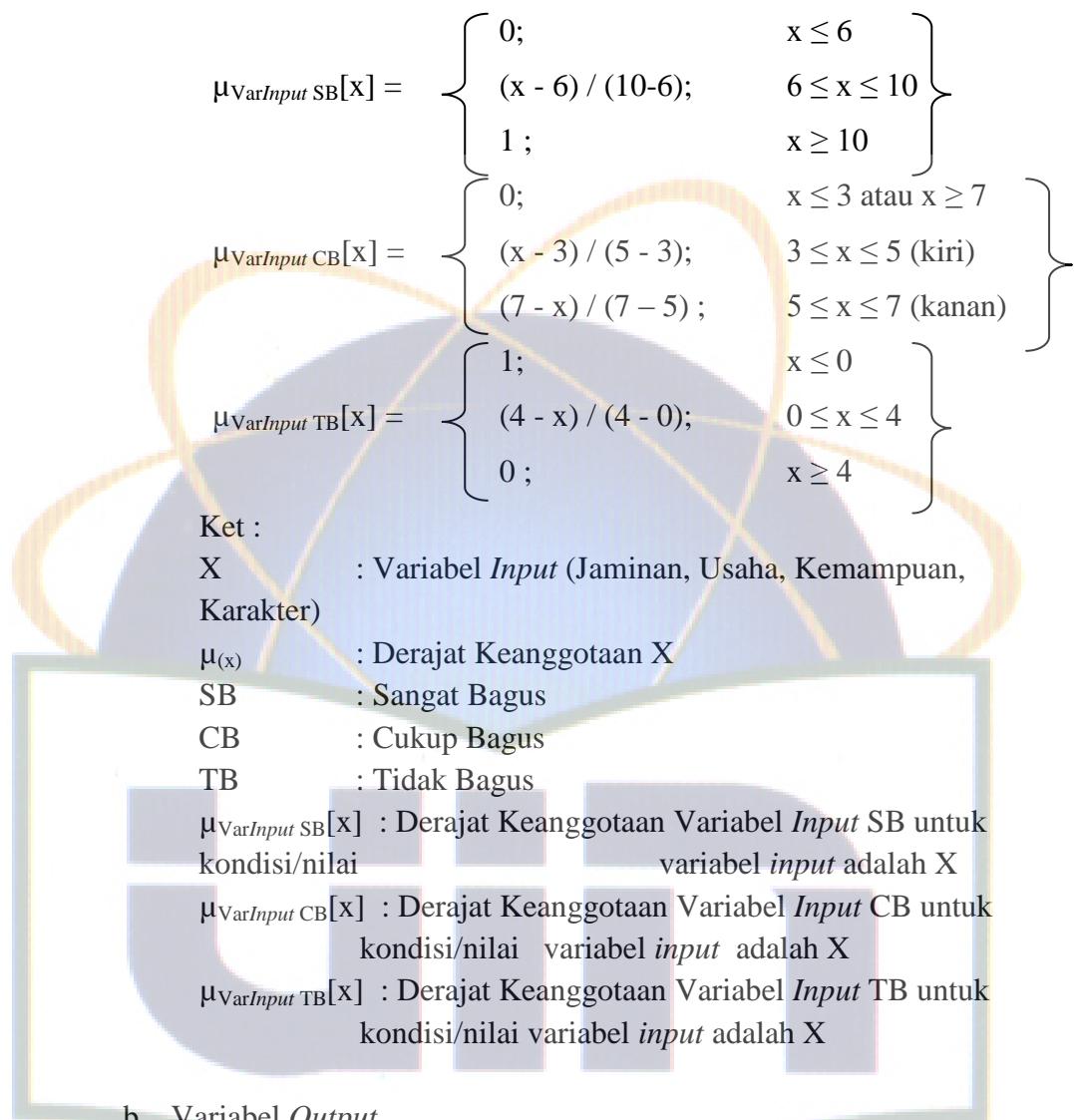
Variabel Input	Fuzzy	Nilai	Range
1. Jaminan	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6 - 10 3 - 7 0 - 4
2. Usaha	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	3 - 10 3 - 7 0 - 4
3. Kemampuan	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	3 - 10 3 - 7 0 - 4
4. Karakter	F	a. Sangat Bagus (SB) b. Cukup Bagus (CB) c. Tidak Bagus (TB)	6 - 10 3 - 7 0 - 4

Fungsi Keanggotaan untuk variabel *input* terlihat pada gambar

4.3 berikut ini:



Gambar 4.3 Fungsi Keanggotaan Variabel *Input*



### b. Variabel *Output*

Variabel *output* terdiri dari satu varibel, yaitu evaluasi penilaian, yang terdiri dari lima himpunan *fuzzy*.

Variabel *output* merupakan hasil evaluasi penilaian berupa skor penilaian kelayakan. nilai maksimal yang dihasilkan apabila semua kriteria dari variabel *input* memiliki kondisi Sangat Bagus adalah sebesar 10, sedangkan apabila kondisi untuk semua kriteria adalah sangat Tidak Bagus, nilai yang dihasilkan adalah 0.

Selang nilai tersebut dikelompokkan kedalam lima kategori kelayakan, pembagian selang untuk masing-masing kelompok dilakukan dengan menggunakan persentase yaitu 10% dari nilai minimum ditetapkan sebagai kategori sangat tidak layak, yang kemudian dilanjutkan 30% untuk kategori kurang tidak layak, 20% untuk kategori layak dengan resiko rendah, dan 10% untuk kategori sangat layak. Penetapan selang tersebut dilakukan oleh pihak BPRS WAKALUMI. Untuk klasifikasi nilai hasil evaluasi lebih jelasnya dapat dilihat pada tabel 4.4 berikut ini.

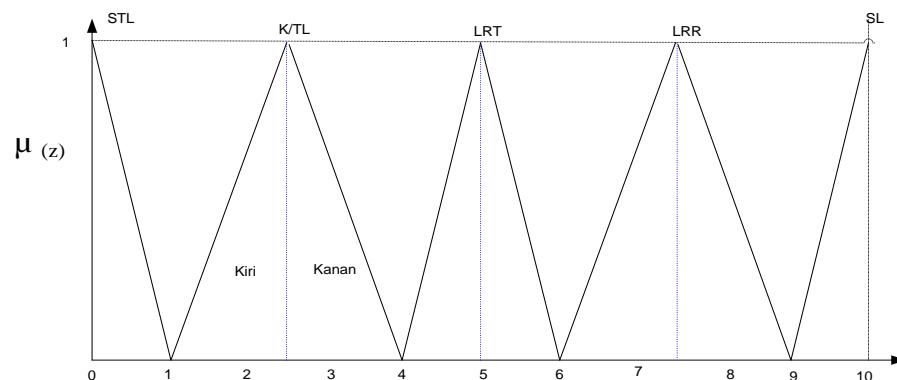
Tabel 4.4 Klasifikasi Nilai Hasil Evaluasi Penilaian

Variabel	Fuzzy	Nilai	Range
Evaluasi Penilaian (Z)	F	a. Sangat Layak (SL) b. Layak dengan Resiko Rendah (LRR) c. Layak dengan Resiko Tinggi (LRT) d. Kurang/ Tidak Layak (K/TL) e. Sangat Tidak Layak (STL)	9 < Z ≤ 10 6 < Z ≤ 9 4 < Z ≤ 6 1 < Z ≤ 4 0 < Z ≤ 1

Sumber: Data Sekunder

Fungsi Keanggotaan untuk variabel *output* terlihat pada Gambar 4.4

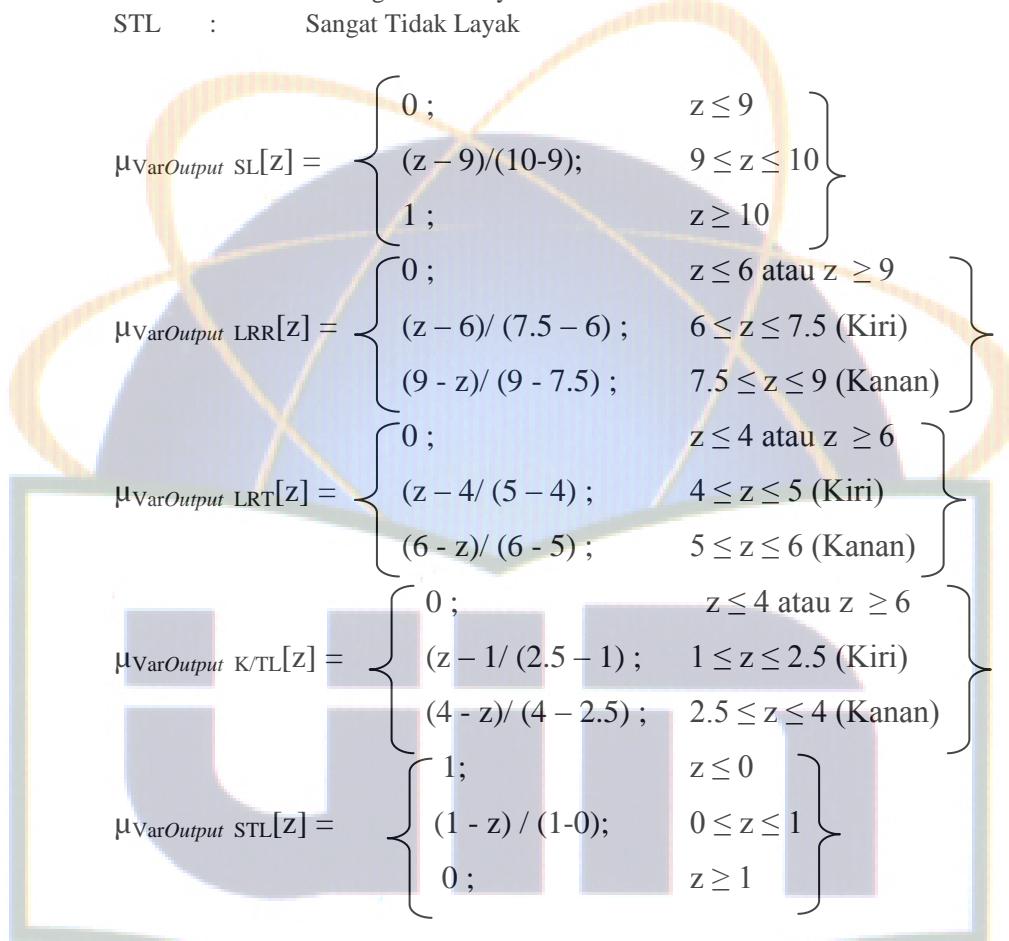
berikut:



Gambar 4.4 Fungsi Keanggotaan Variabel *Output*

Ket:

- SL : Sangat Layak
- LRR : Layak dengan Resiko Rendah
- LRT : Layak dengan Resiko Tinggi
- K/TL : Kurang/ Tidak Layak
- STL : Sangat Tidak Layak



#### 4.2.2.3 Membuat Aturan Fuzzy

Sistem penilaian kelayakan pembiayaan ini dibangun oleh 81 aturan-aturan *fuzzy* berdasarkan proses bisnis di BPRS WAKALUMI. 81 aturan *fuzzy* tersebut berasal dari 3<sup>4</sup>. Tiga merupakan Jumlah nilai yang digunakan oleh setiap kriteria atau variabel, sedangkan empat merupakan jumlah variabel yang terlibat dalam proses penilaian kelayakan pembiayaan. Aturan-aturan *fuzzy* dapat dilihat pada Table 4.5 berikut ini:

Tabel 4.5 Aturan *Fuzzy* Berdasarkan Proses Bisnis Di BPRS WAKALUMI

NO	<b>VARIABEL INPUT</b>				<b>VARIABEL OUTPUT</b>
	Jaminan	Usaha	Kemampuan	Karakter	Evaluasi Penilaian
1	SB	SB	SB	SB	SL
2	SB	SB	SB	CB	SL
3	SB	SB	SB	TB	LRR
4	SB	SB	CB	SB	SL
5	SB	SB	CB	CB	LRR
6	SB	SB	CB	TB	LRT
7	SB	SB	TB	SB	LRR
8	SB	SB	TB	CB	LRT
9	SB	SB	TB	CB	LRT
10	SB	CB	SB	CB	LRR
11	SB	CB	SB	CB	LRR
12	SB	CB	SB	TB	LRR
13	SB	CB	CB	SB	LRR
14	SB	CB	CB	CB	LRR
15	SB	CB	CB	TB	LRT
16	SB	CB	TB	SB	LRT
17	SB	CB	TB	CB	K/TL
18	SB	CB	TB	TB	K/TL
19	SB	TB	SB	SB	LRR
20	SB	TB	SB	CB	LRT
21	SB	TB	SB	TB	K/TL
22	SB	TB	CB	SB	LRT
23	SB	TB	CB	CB	LRT
24	SB	TB	CB	TB	K/TL
25	SB	TB	TB	SB	K/TL
26	SB	TB	TB	CB	K/TL
27	SB	TB	TB	TB	K/TL
28	CB	SB	SB	SB	SL
29	CB	SB	SB	CB	SL
30	CB	SB	SB	TB	LRR
31	CB	SB	CB	SB	LRR
32	CB	SB	CB	CB	LRR

33	CB	SB	CB	TB	LRT
34	CB	SB	TB	SB	LRT
35	CB	SB	TB	CB	LRT
36	CB	SB	TB	TB	K/TL
37	CB	CB	SB	SB	LRR
38	CB	CB	SB	CB	LRR
39	CB	CB	SB	TB	LRT
40	CB	CB	CB	SB	LRR
41	CB	CB	CB	CB	LRR
42	CB	CB	CB	TB	LRT
43	CB	CB	TB	SB	LRT
44	CB	CB	TB	CB	LRT
45	CB	CB	TB	TB	K/TL
46	CB	TB	SB	SB	K/TL
47	CB	TB	SB	CB	K/TL
48	CB	TB	SB	TB	K/TL
49	CB	TB	CB	SB	K/TL
50	CB	TB	CB	CB	K/TL
51	CB	TB	CB	TB	K/TL
52	CB	TB	TB	SB	K/TL
53	CB	TB	TB	CB	K/TL
54	CB	TB	TB	TB	K/TL
55	TB	SB	SB	SB	K/TL
56	TB	SB	SB	CB	K/TL
57	TB	SB	SB	TB	K/TL
58	TB	SB	CB	SB	K/TL
59	TB	SB	CB	CB	K/TL
60	TB	SB	CB	TB	K/TL
61	TB	SB	TB	SB	K/TL
62	TB	SB	TB	CB	K/TL
63	TB	SB	TB	TB	STL
64	TB	CB	SB	SB	K/TL
65	TB	CB	SB	CB	K/TL
66	TB	CB	SB	TB	K/TL
67	TB	CB	CB	SB	K/TL
68	TB	CB	CB	CB	K/TL
69	TB	CB	CB	TB	K/TL
70	TB	CB	TB	SB	K/TL
71	TB	CB	TB	CB	STL

72	TB	CB	TB	TB	STL
73	TB	TB	SB	SB	K/TL
74	TB	TB	SB	CB	K/TL
75	TB	TB	SB	TB	STL
76	TB	TB	CB	SB	K/TL
77	TB	TB	CB	CB	STL
78	TB	TB	CB	TB	STL
79	TB	TB	TB	SB	STL
80	TB	TB	TB	CB	STL
81	TB	TB	TB	TB	STL

Sumber: Data Sekunder

Untuk memudahkan penulisan menampilkan aturan-aturan tersebut dalam bentuk tabel. Salah satu contoh penggunaan aturan *if-then* untuk aturan pertama adalah:

[R1] *IF* Jaminan SB And Usaha SB And Kemampuan SB And Karakter SB *THEN* Evaluasi Penilaian SL.

#### 4.2.2.4 Menentukan Metode Defuzzy Untuk Setiap Variabel

Setelah derajat keanggotaan evaluasi penilaian dari masing-masing aturan *fuzzy* terbentuk, selanjutnya dihitung nilai hasil evaluasi penilaian.

Karena pada penelitian ini penulis menggunakan metode Tsukamoto, maka untuk menghitung nilai hasil evaluasi penilaian yang direkomendasikan menggunakan rumus sebagai berikut:

$$Z = \frac{(\alpha_1 * Z_1) + (\alpha_2 * Z_2) + (\alpha_3 * Z_3) + \dots + (\alpha_{81} * Z_{81})}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_{81}}$$

Ket:

- $Z$  rekomendasi nilai hasil evaluasi penilaian.
- $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{81}$  = Derajat keanggotaan evaluasi penilaian dari tiap-tiap aturan *Fuzzy*.
- $Z_1, Z_2, Z_3, Z_4, \dots, Z_{81}$  = Nilai hasil evaluasi penilaian berdasarkan model *Fuzzy* keanggotaan evaluasi penilaian dari tiap-tiap aturan *Fuzzy*

Adapun alasan dipergunakan metode Tsukamoto adalah karena metode ini memberikan *output* yang lebih besar dibandingkan metode lainnya, tetapi tetap dalam kisaran yang wajar. Hal ini lebih baik, karena sifat BPRS Wakalumi yang lebih fleksibel dalam memberikan layanan pembiayaan kepada nasabah, namun tetap memperhitungkan ketelitian, mengingat dalam kenyataannya banyak pengusaha kecil menengah yang tidak layak masuk daftar mendapat layanan pembiayaan atau pinjaman dari bank.

#### 4.2.2.5 Menjalankan Simulasi Sistem dan Pengujian SPK Kelayakan UKM

Langkah terakhir dari *design fuzzy* adalah menjalankan simulasi sistem dan pengujian terhadap beberapa variabel *input* yang berbeda.

Berikut ini adalah salah satu contoh *input* kondisi dari seorang Calon Debitur yang mengajukan permohonan pembiayaan di BPRS Wakalumi. Sistem ini mencoba menghitung hasil nilai evaluasi penilaian kelayakan pembiayaan dari Calon Debitur, jenis Usaha Pedagang Daging dengan kondisi sebagai berikut. Lihat tabel 4.6.

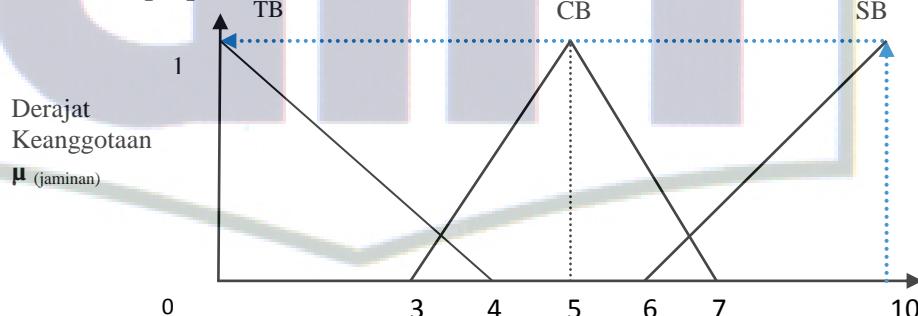
Tabel 4.6 Kombinasi Masukan Calon Debitur 1 – Pedagang Daging

Variabel Input (x)	Parameter	Kondisi	Bobot
a. Jaminan = 10	1. Nilai Jaminan	Sangat Mendukung	<b>10.0</b>
b. Usaha = 6.75	1. Modal Usaha	Sendiri	2.0
	2. Omzet	Sedang	1.0
	3. Usaha	Cukup strategis	0.75
	4. Jenis tempat Usaha	Warung/Tenda	0.5

	5. Lama Usaha	Lama	1.0
	6. Persaingan	Sedang	0.5
	7. Kepemilikan	Milik sendiri	0.75
	8. Jumlah tenaga kerja	Sedikit	0.25
	<b>Jumlah</b>		<b>6.75</b>
c. Kemampuan = 5.25	1. Penghasilan perbulan	Sedang	1.25
	2. Pengeluaran perbulan	Sedang	1.25
	3. Kepemilikan tempat tinggal	Kontrak	0.5
	4. Kondisi tempat tinggal	Permanen	1.5
	5. Jumlah tanggungan	Sedang	0.75
	<b>Jumlah</b>		<b>5.25</b>
d. Karakter = 6.25	1. Perilaku sehari-hari	Baik	2.5
	2. Penampilan	Cukup Baik	1.25
	3. Interaksi bermasyarakat	Cukup Baik	1.25
	4. Ketaatan beribadah	Cukup Baik	1.25
	<b>Jumlah</b>		<b>6.25</b>

Variabel fuzzy yang dimodelkan adalah sebagai berikut:

a. **Jaminan [10]**



Gambar 4.5 Fungsi Keanggotaan Variabel Jaminan pada contoh Tabel 4.6

Fungsi Keanggotaan dimana range Jaminan  $SB = 6 - 10$ , yaitu :

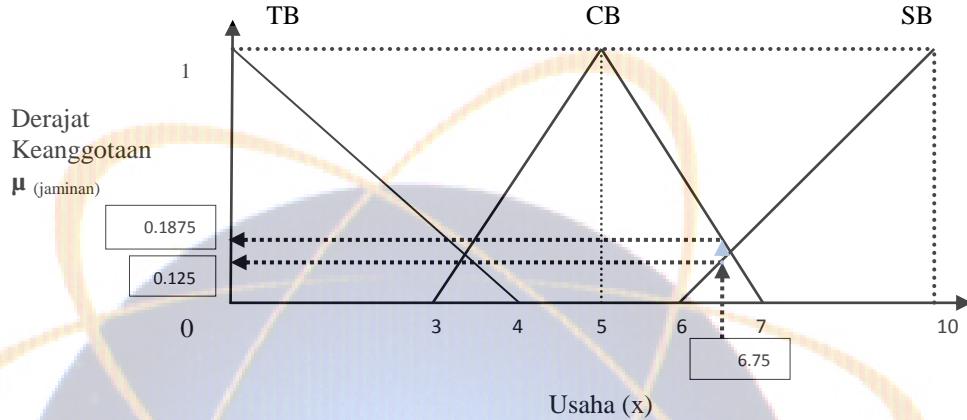
$$\mu_{\text{VarInput}} \text{SB}[x] = \begin{cases} 0; & x \leq 6 \\ (x - 6) / (10-6); & 6 \leq x \leq 10 \\ 1; & x \geq 10 \end{cases}$$

Derajat Keanggotaan jika nilai Jaminan atau  $x = 10$ , yaitu :

$$\mu_{\text{Jaminan}} \text{SB}[10] = 1$$

karena  $x \geq 10$

**b. Usaha [6.75]**



Gambar 4.6 Fungsi Keanggotaan Variabel Usaha pada contoh Tabel 4.6

Fungsi Keanggotaan dimana range Usaha  $SB = 6 - 10$ , dan Usaha  $CB = 3 - 7$ , yaitu :

$$\mu_{VarInput\ SB}[x] = \begin{cases} 0; & x \leq 6 \\ (x - 6) / (10 - 6); & 6 \leq x \leq 10 \\ 1; & x \geq 10 \end{cases}$$

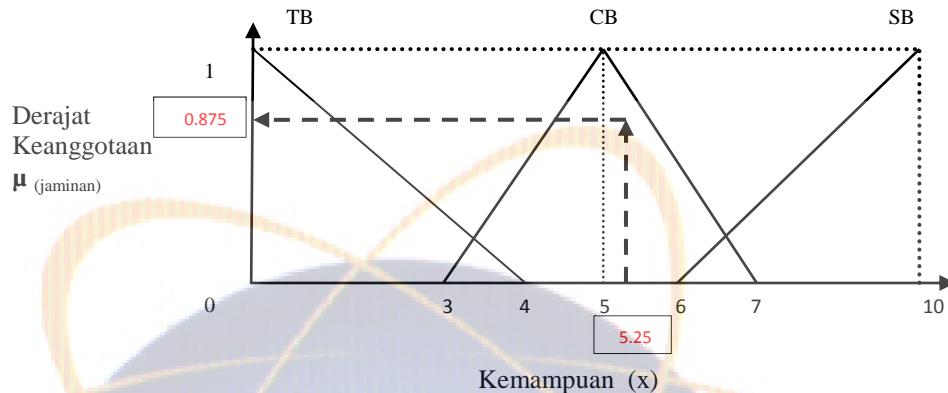
$$\mu_{VarInput\ CB}[x] = \begin{cases} 0; & x \leq 3 \text{ atau } x \geq 7 \\ (x - 3) / (5 - 3); & 3 \leq x \leq 5 \text{ (kiri)} \\ (7 - x) / (7 - 5); & 5 \leq x \leq 7 \text{ (Kanan)} \end{cases}$$

Derajat Keanggotaan jika nilai Usaha atau  $x = 6.75$ , yaitu :

$$\mu_{Usaha\ SB}[6.75] = (6.75 - 6) / 4 \quad \text{karena } 6 \leq x \leq 10 \\ = \mathbf{0.1875}$$

$$\mu_{Usaha\ CB}[6.75] = (7 - 6.75) / 2 \quad \text{karena } 5 \leq x \leq 7 \text{ (Kanan)} \\ = \mathbf{0.125}$$

### c. Kemampuan [5.25]



Gambar 4.7 Fungsi Keanggotaan Variabel Kemampuan pada Contoh Tabel 4.6

Fungsi Keanggotaan dimana range Kemampuan CB = 3 – 7, yaitu :

$$\mu_{VarInput\ CB}[x] = \begin{cases} 0; & x \leq 3 \text{ atau } x \geq 7 \\ (x-3) / (5-3); & 3 \leq x \leq 5 \text{ (kiri)} \\ (7-x) / (7-5); & 5 \leq x \leq 7 \text{ (Kanan)} \end{cases}$$

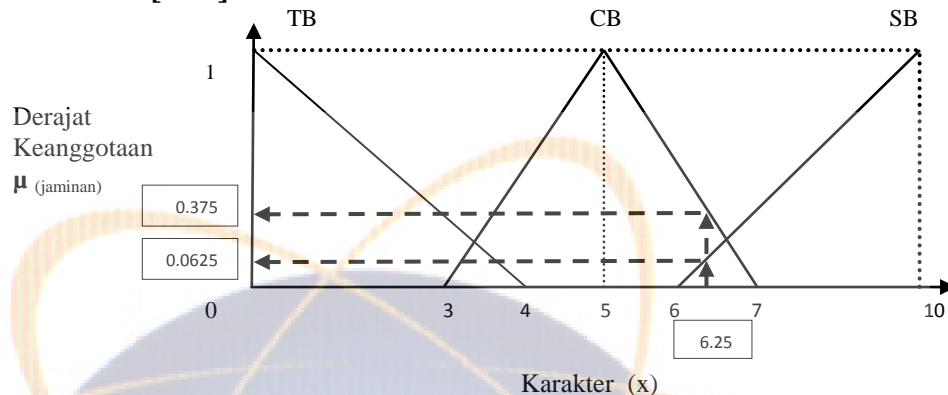
Derajat Keanggotaan jika nilai Usaha atau  $x = 4.5$ , yaitu :

$$\begin{aligned} \mu_{Kemampuan\ CB}[5.25] &= (7-5.25)/2 \quad \text{karena } 5 \leq x \leq 7 \text{ (Kanan)} \\ &= 0.875 \end{aligned}$$

Untuk mencari derajat keanggotaan usaha menggunakan rumus

$\mu_{Kemampuan\ CB}$  sebelah kanan, karena 5.25 berada di posisi sebelah kiri dalam kurva segitiga.

**d. Karakter [6.25]**



Gambar 4.8 Fungsi Keanggotaan Variabel Karakter pada Contoh

Tabel 4.6

Fungsi Keanggotaan dimana Karakter  $SB = 6 - 10$ , dan Karakter  $CB = 3 - 7$ , yaitu :

$$\mu_{VarInput\ SB}[x] = \begin{cases} 0; & x \leq 6 \\ (x-6) / (10-6); & 6 \leq x \leq 10 \\ 1; & x \geq 10 \end{cases}$$

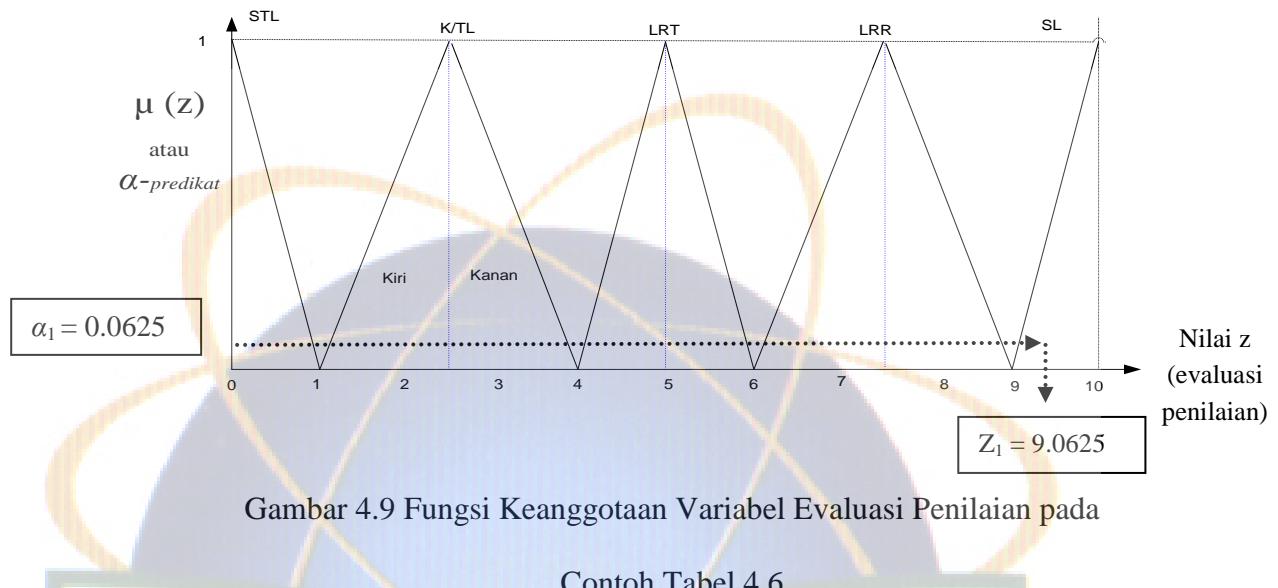
$$\mu_{VarInput\ CB}[x] = \begin{cases} 0; & x \leq 3 \text{ atau } x \geq 7 \\ (x-3) / (5-3); & 3 \leq x \leq 5 \text{ (kiri)} \\ (7-x) / (7-5); & 5 \leq x \leq 7 \text{ (Kanan)} \end{cases}$$

Derajat Keanggotaan jika nilai Karakter atau  $x = 6.25$ , yaitu :

$$\mu_{Usaha\ SB}[6.25] = (6.25 - 6) / 4 \quad \text{karena } 6 \leq x \leq 10 \\ = 0.0625$$

$$\mu_{Usaha\ CB}[6.25] = (7 - 6.25) / 2 \quad \text{karena } 5 \leq x \leq 7 \text{ (kanan)} \\ = 0.375$$

### e. Evaluasi Penilaian



Contoh Tabel 4.6

**Contoh Tabel 4.6**

$$\mu_{\text{VarOutput SL}}[z] = \begin{cases} 0; & z \leq 9 \\ (z-9) / (10-9); & 9 \leq z \leq 10 \\ 1; & z \geq 10 \end{cases}$$

$$\mu_{\text{VarOutput LRR}}[z] = \begin{cases} 0; & z \leq 6 \text{ atau } z \geq 9 \\ (z-6) / (7.5-6); & 6 \leq z \leq 7.5 \text{ (kiri)} \\ (9-z) / (9-7.5); & 7.5 \leq z \leq 9 \text{ (Kanan)} \end{cases}$$

$$\mu_{\text{VarOutput LRT}}[z] = \begin{cases} 0; & z \leq 4 \text{ atau } z \geq 6 \\ (z-4) / (5-4); & 4 \leq z \leq 5 \text{ (kiri)} \\ (6-z) / (6-5); & 7.5 \leq z \leq 9 \text{ (Kanan)} \end{cases}$$

$$\mu_{\text{VarOutput K/TL}}[z] = \begin{cases} 0; & z \leq 1 \text{ atau } z \geq 4 \\ (z-1) / (2.5-1); & 1 \leq z \leq 2.5 \text{ (kiri)} \\ (4-z) / (4-2.5); & 2.5 \leq z \leq 4 \text{ (Kanan)} \end{cases}$$

$$\mu_{\text{VarOutput STL}}[z] = \begin{cases} 1; & z \leq 0 \\ (1-z) / (1-0); & 0 \leq z \leq 1 \\ 0; & z \geq 1 \end{cases}$$

Sebelum mencari nilai z untuk setiap aturan terlebih dahulu ditentukan aturan-aturan yang digunakan, karena dari delapan puluh satu (81) jumlah aturan yang ada [lihat Tabel.1/ Aturan-aturan Fuzzy], tidak semua digunakan, tergantung dari kondisi dari setiap variabel yang di *input*, pada contoh kasus ini. Aturan yang digunakan hanya empat (4), yaitu:

[R4] *IF Jaminan SB And Usaha SB And Kemampuan CB And Karakter SB THEN Evaluasi penilaian SL;*

[R5] *IF Jaminan SB And Usaha SB And Kemampuan CB And Karakter CB THEN Evaluasi penilaian LRR;*

[R13] *IF Jaminan SB And Usaha CB And Kemampuan CB And Karakter SB THEN Evaluasi penilaian LRR;*

[R14] *IF Jaminan SB And Usaha CB And Kemampuan CB And Karakter CB THEN Evaluasi penilaian LRR;*

Setelah semua aturan terbentuk, selanjutnya mencari nilai z untuk setiap aturan dengan menggunakan fungsi *MIN* pada aplikasi fungsi implikasinya.

[R4] *IF Jaminan SB And Usaha SB And Kemampuan CB And Karakter SB THEN Evaluasi penilaian SL;*

$$\begin{aligned}
 \alpha_1 &= \mu_{JaminanSB} \cap \mu_{UsahaSB} \cap \mu_{KemampuanCB} \cap \mu_{KarakterSB} \\
 &= \min (\mu_{JaminanSB}[10], \mu_{UsahaSB}[6.5], \mu_{KemampuanCB}[4.5], \\
 &\quad \mu_{KarakterSB}[6.25]) \\
 &= \min (1 ; 0.1875 ; 0.875 ; 0.0625) \\
 &= \mathbf{0.0625}
 \end{aligned}$$

Berdasarkan Himpunan Evaluasi penilaian SL, Maka dapat dicari nilai  $z_1$  untuk [R4],

$$(z_1 - 9)/(10-9) = 0.0625$$

$$z_1 - 9 = 0.0625 * 1$$

$$z_1 = 0.0625 + 9$$

$$z_1 = 9.0625$$

$$\rightarrow z_1 = \mathbf{9.0625}$$

[R5] *IF Jaminan SB And Usaha SB And Kemampuan CB And Karakter CB THEN Evaluasi penilaian LRR;*

$$\alpha_2 = \mu_{JaminanSB} \cap \mu_{UsahaSB} \cap \mu_{KemampuanCB} \cap \mu_{KarakterSB} \\ = \min (\mu_{JaminanSB}[10], \mu_{UsahaSB}[6.5], \mu_{KemampuanCB}[4.5],$$

$$\mu_{KarakterCB}[6.25])$$

$$= \min (1 ; 0.1875 ; 0.875 ; 0.375)$$

$$= \mathbf{0.1875}$$

Berdasarkan Himpunan Evaluasi penilaian LRR, maka dapat dicari nilai  $z_2$  untuk [R5],

$$(9-z_2)/(9-7.5) = 0.1875$$

$$9-z_2 = 0.1875 * 1.5$$

$$-z_2 = 0.28125 - 9$$

$$z_2 = 8.71875$$

$$\rightarrow z_2 = \mathbf{8.71875}$$

[R13] *IF Jaminan SB And Usaha CB And Kemampuan CB And Karakter SB THEN Evaluasi penilaian LRR;*

$$\alpha_3 = \mu_{JaminanSB} \cap \mu_{UsahaCB} \cap \mu_{KemampuanCB} \cap \mu_{KarakterSB}$$

$$\begin{aligned}
 &= \min (\mu_{\text{JaminanSB}}[10], \mu_{\text{UsahaCB}}[6.5], \mu_{\text{KemampuanCB}}[4.5], \\
 &\quad \mu_{\text{KarakterSB}}[6.25]) \\
 &= \min (1 ; 0.125 ; 0.875 ; 0.0625) \\
 &= \mathbf{0.0625}
 \end{aligned}$$

Berdasarkan Himpunan Evaluasi penilaian LRR, maka dapat dicari nilai  $z_3$  untuk [R13],

$$(9-z_3)/(9-7.5) = 0.0625$$

$$9-z_3 = 0.0625 * 1.5$$

$$-z_3 = 0.09375 - 9$$

$$z_3 = 8.90625$$

$$\rightarrow z_3 = \mathbf{8.90625}$$

[R14] *IF Jaminan SB And Usaha CB And Kemampuan CB And Karakter CB THEN Evaluasi penilaian LRR;*

$$\begin{aligned}
 \alpha_4 &= \mu_{\text{JaminanSB}} \cap \mu_{\text{UsahaCB}} \cap \mu_{\text{KemampuanCB}} \cap \mu_{\text{KarakterCB}} \\
 &= \min (\mu_{\text{JaminanSB}}[10], \mu_{\text{UsahaCB}}[6.5], \mu_{\text{KemampuanCB}}[4.5], \\
 &\quad \mu_{\text{KarakterCB}}[6.25]) \\
 &= \min (1 ; 0.125 ; 0.875 ; 0.375) \\
 &= \mathbf{0.125}
 \end{aligned}$$

Berdasarkan Himpunan Evaluasi penilaian LRR, maka dapat dicari nilai  $z_4$  untuk [R14],

$$(9-z_4)/(9-7.5) = 0.125$$

$$9-z_4 = 0.125 * 1.5$$

$$-z_4 = 0.1875 - 9$$

$$z_4 = 8.8125 \quad \rightarrow \quad z_4 = \mathbf{8.8125}$$

Setelah mendapatkan nilai  $z_1, z_2, z_3, z_4$  untuk setiap aturan maka dicari nilai  $z$ , yaitu :

$$\begin{aligned}
 Z &= \frac{(\alpha_1 * z_1) + (\alpha_2 * z_2) + (\alpha_3 * z_3) + (\alpha_4 * z_4)}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4} \\
 &= \frac{(0.0625 * 9.0625) + (0.1875 * 8.71875) + (0.0625 * 8.90625) + (0.125 * 8.8125)}{0.0625 + 0.1875 + 0.0625 + 0.125} \\
 &= \frac{0.5664063 + 1.6347656 + 0.5566406 + 1.105625}{0.4375} \\
 &= \frac{3.859375}{0.4375}
 \end{aligned}$$

$Z = 8.821428571428571$

Jadi nilai *Fuzzy* hasil perhitungan *Evaluasi* sebesar **8.821428571428571**, dengan keterangan kelayakan adalah **Layak dengan Resiko Rendah** [lihat tabel 4.4 Klasifikasi Nilai Hasil Evaluasi Penilaian]

Berikut Tabel Simulasi dengan berbagai variasi masukan dan keluaran. Tabel simulasi ini diperlukan guna mengetahui apakah proses pengolahan data *fuzzy* sudah sesuai dengan model yang diterapkan dalam perancangan sistem ini.



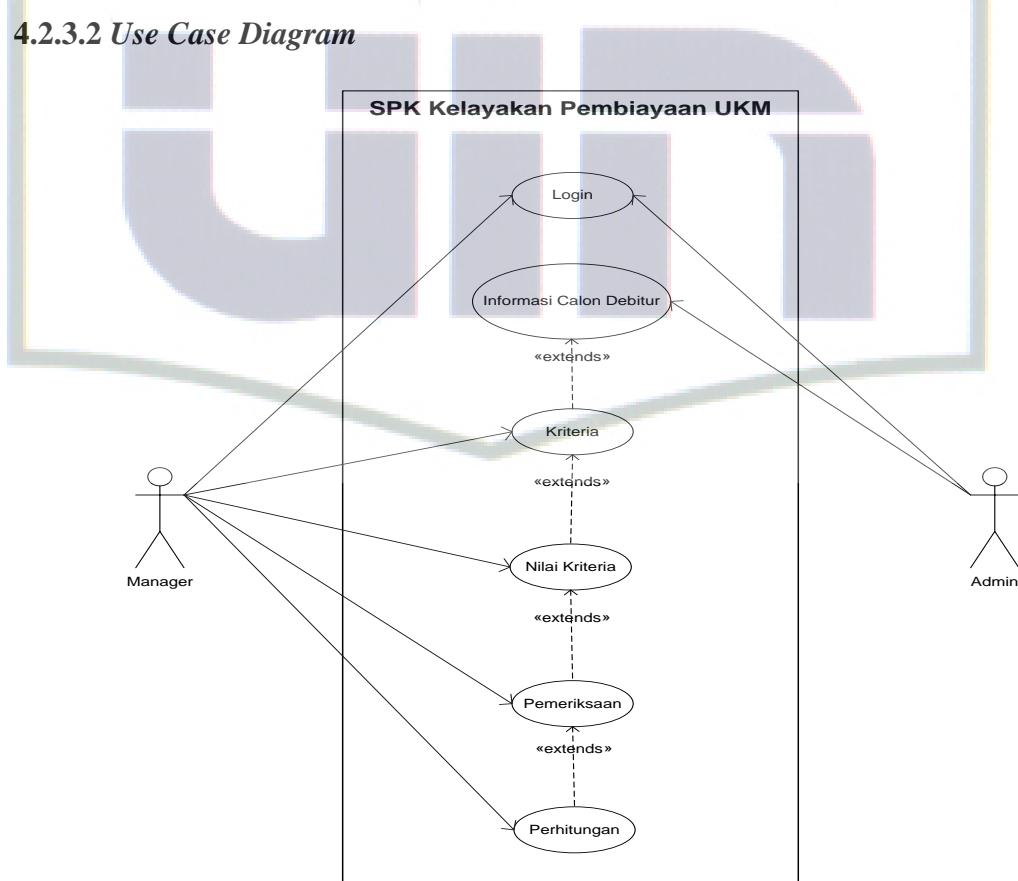
### 4.2.3 Rancangan Spesifikasi SPK

#### 4.2.3.1 Identifikasi Actor

Tabel 4.8 Identifikasi *actor*

No	Actor	Description
1	Manager	Memiliki wewenang untuk menentukan kriteria yang berlaku dan melakukan perhitungan dengan logika <i>Fuzzy</i>
2	Admin	Memiliki wewenang untuk memasukkan informasi biodata Calon Debitur yang akan mengajukan pembiayaan.

#### 4.2.3.2 Use Case Diagram



Gambar 4.10 – *Use Case Diagram* Sistem Penunjang Keputusan yang Diajukan

#### 4.2.3.2.1 Identifikasi Use Case Diagram

Tabel 4.9 - Identifikasi Use Case Diagram

No	Use Case Name	Deskripsi	Actor
1	Login	Use Case ini digunakan untuk masuk ke dalam aplikasi SPK pembiayaan kelayakan UKM	Manager dan Admin
2	Kriteria	Use Case ini digunakan untuk mengisi nama kriteria yang dibandingkan	Manager
3	Nilai Kriteria	Use Case ini digunakan untuk mengisi nilai kriteria terhadap Calon Debitur	Manager
4	Pemeriksaan	Use Case ini digunakan untuk menghitung nilai dengan menggunakan logika fuzzy	Manager
5	Perhitungan	Use Case ini digunakan untuk melihat hasil perhitungan nilai yang telah dihasilkan	Manager
6	Informasi Calon Debitur	Use Case ini digunakan untuk mengisi informasi biodata pemohon yang berkaitan dengan pinjaman	Admin

#### 4.2.3.2.2 Narasi Use Case

##### 1. Use Case Login

Tabel 4.10 – Narasi Use Case Login

Use Case Name	Login
Actor	User (Manager dan Admin).
Description	Use Case ini digunakan Manager dan admin untuk masuk kedalam system.
References	Gambar 4.10

	<i>Actor Action</i>	<i>System Response</i>
<i>Typical Course of Events</i>	1: Input <i>username</i> dan <i>password</i> .	2: validasi <i>username</i> , <i>password</i> 3: Tampilan halaman utama
<i>Alternative Course</i>	2. Jika <i>password</i> dan <i>username</i> ditolak, maka sistem akan meminta <i>user</i> memasukan <i>username</i> dan <i>password</i> yang benar.	
<i>Post Condition</i>	Melakukan pilihan menu pada masing – masing status <i>user</i> .	

## 2. Use Case Kriteria

Tabel 4.11 – Narasi Use Case Kriteria

<i>Use Case Name</i>	<i>Kriteria</i>	
<i>Actor (s)</i>	Manager	
<i>Description</i>	<i>Use Case</i> ini digunakan Manager untuk mengisi nama parameter kriteria yang berlaku.	
<i>References</i>	Gambar 4.10	
<i>Typical Course of Events</i>	<i>Actor Action</i>	<i>System Response</i>
	1 = pilih menu kriteria 3 = klik add new kriteria 5 = input data kriteria 6 = pilih save	2 = menampilkan halaman kriteria 4= menampilkan form tambah kriteria 7= simpan data kriteria 8= menampilkan halaman kriteria
<i>Alternative Course</i>	6. Jika data belum lengkap, maka sistem akan meminta user melengkapi data. 8. jika ingin hapus kriteria, maka actor memilih kriteria yang akan dihapus 8. jika ingin edit kriteria, maka actor memilih kriteria yang akan di edit.	
<i>Post Conditions</i>	Actor melakukan penyimpanan	

### 3. Use Case Nilai Kriteria

Tabel 4.12 – Narasi Use Case Nilai Kriteria

<b>Use Case Name</b>	<b>Nilai Kriteria</b>	
<b>Actor (s)</b>	Manager	
<b>Description</b>	<i>Use Case</i> ini digunakan Manager untuk mengisi nilai kriteria yang akan dihitung <i>fuzzy</i> .	
<b>References</b>	Gambar 4.10	
<b>Typical Course of Events</b>	<b>Actor Action</b> 1 = memilih menu nilai kriteria 3 = klik new nilai kriteria 5 = input data nilai kriteria 6= Pilih save	<b>System Response</b> 2 = menampilkan form nilai kriteria 4 = menampilkan form tambah nilai kriteria 7 = simpan nilai kriteria 8= menampilkan halaman nilai kriteria
<b>Alternative Course</b>	6 Jika data belum lengkap, maka sistem akan meminta user melengkapi data 8 jika ingin hapus data nilai, maka actor memilih data yang akan dihapus pada halaman nilai kriteria, kemudian klik delete. 8 jika ingin edit data nilai, maka actor memilih data yang akan di edit pada halama nilai kriteria, kemudian klik edit	
<b>Post Conditions</b>	Actor melakukan penyimpanan	

### 7 Use Case Pemeriksaan

Tabel 4.13 – Narasi Use Case Pemeriksaan

<b>Use Case Name</b>	<b>Pemeriksaan</b>
<b>Actor (s)</b>	Manager
<b>Description</b>	<i>Use Case</i> ini digunakan Manager untuk memeriksa data dan informasi yang didapat dari hasil <i>survey</i>

<b>References</b>	Gambar 4.10	
<b>Typical Course of Events</b>	<b>Actor Action</b> 1 = pilih menu pemeriksaan 3 = pilih nama calon 5 = input data kelayakan 6 = pilih tombol periksa 8= print	<b>System Response</b> 2 = menampilkan menu pemeriksaan 4 = menampilkan nama calon pemohon 7 = menampilkan hasil perhitungan
<b>Alternative Course</b>	5 jika ingin mengulang input data kelayakan, maka actor memilih tombol reset. 6 jika data belum lengkap, maka sistem akan meminta user untuk melengkapi data.	
<b>Post Conditions</b>	Actor melakukan penyimpanan dan perhitungan	

### 8 Use Case Perhitungan

Tabel 4.14 – Narasi Use Case Perhitungan

<b>Use Case Name</b>	Perhitungan	
<b>Actor (s)</b>	Manager	
<b>Description</b>	<i>Use Case</i> ini digunakan Manager untuk melihat dan menghitung hasil Fuzzy	
<b>References</b>	Gambar 4.10	
<b>Typical Course of Events</b>	<b>Actor Action</b> 1 = pilih menu perhitungan 3 = pilih nama pemohon 5 = print	<b>System Response</b> Step 2 = menampilkan form perhitungan 4 = menampilkan nama pemohon
<b>Alternative Course</b>	-	
<b>Post Conditions</b>	Actor melakukan pencetakan	

## 9 Use Case Informasi Calon Debitur

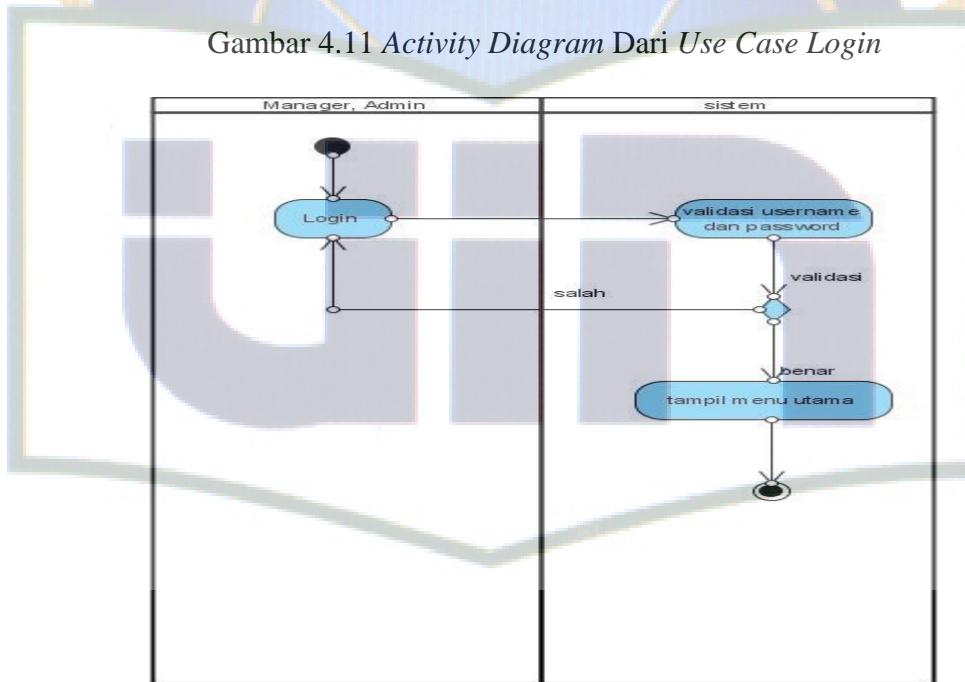
Tabel 4.15 – Narasi Use Case Mengisi Informasi Calon Debitur

<b>Use Case Name</b>	<b>Informasi Calon Debitur</b>	
<b>Actor (s)</b>	<i>Admin</i>	
<b>Description</b>	<i>Use Case ini digunakan Admin untuk mengisi informasi data dari Calon Debitur</i>	
<b>References</b>	Gambar 4.10	
<b>Typical Course of Events</b>	<b>Actor Action</b> 1 = pilih menu Pemohon 3 = klik new pemohon 5 = pilih simpan	<b>System Response</b> 2 = menampilkan form Pemohon 4 = menampilkan form tambah pemohon 6 = melakukan penyimpanan 7 = menampilkan halaman data pemohon
<b>Alternative Course</b>	5 jika data belum lengkap, maka sistem akan meminta user melengkapi data 7 jika ingin mengubah data pemohon, maka actor memilih data pemohon yang akan di ubah, kemudian klik edit. 7 jika ingin menghapus data pemohon, maka actor memilih data pemohon yang akan di hapus, kemudian klik delete.	
<b>Post Conditions</b>	Actor melakukan penyimpanan	

#### 4.2.3.3 Activity Diagram

Pada bagian ini menjelaskan / menampilkan bagaimana sistem dalam melakukan setiap aktivitas-aktivitasnya. Detail kegiatan yang dilakukan setiap *actor* untuk setiap *use case* yang terdapat pada desain sistem penunjang keputusan kelayakan pembiayaan UKM. Berikut ini beberapa *activity diagram* yang terdapat pada sistem penunjang keputusan kelayakan pembiayaan UKM.

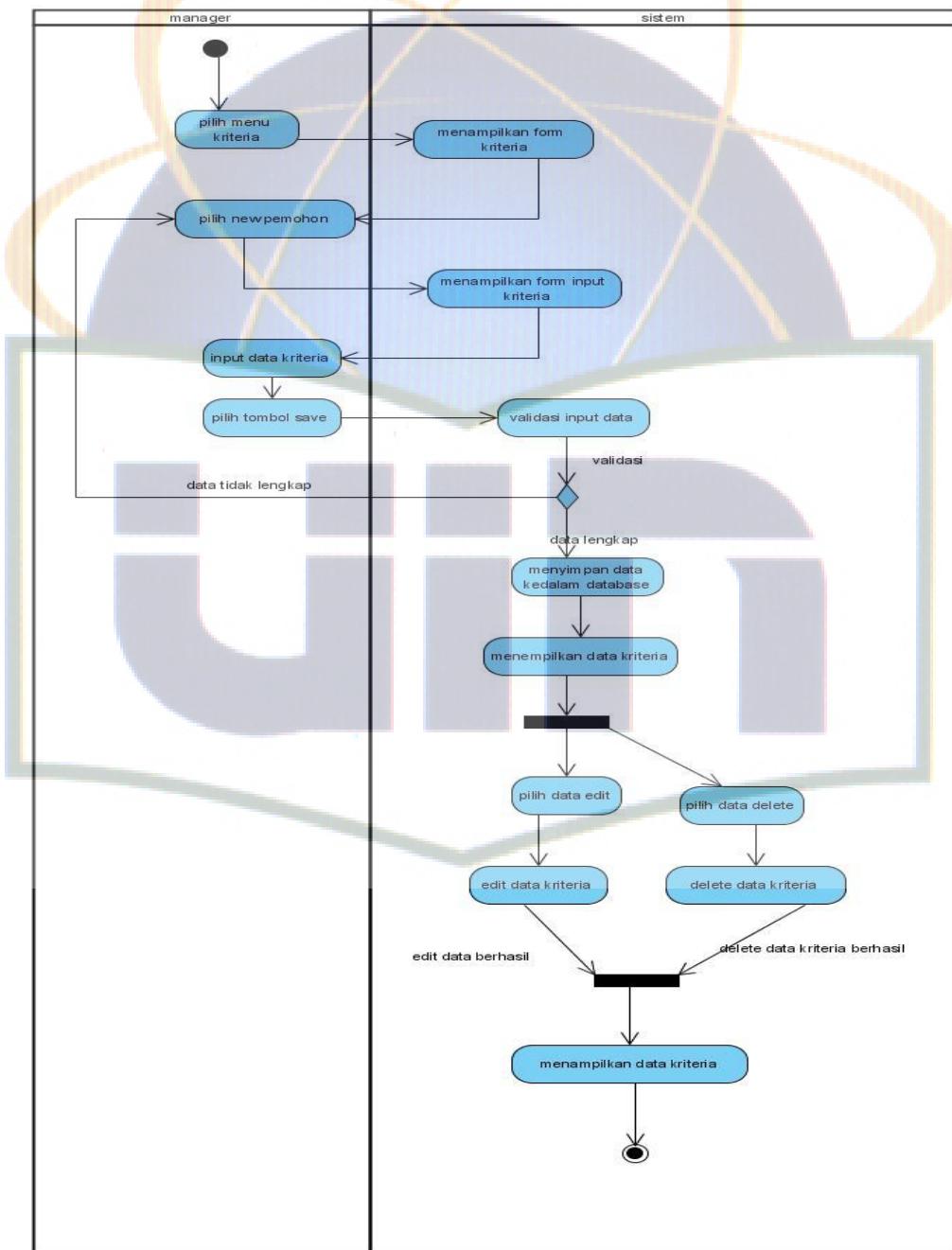
##### 4.2.3.3.1 Activity Diagram Dari Use Case Login



Pada *activity diagram* dari *use case login*, *user* pertama-tama menjalankan aplikasi, setelah aplikasi terbuka tampilan intro aplikasi untuk masuk ke login hanya tekan enter, *form login* akan muncul dan meminta *user* untuk memasukan *username* dan *password*. Jika *username* dan *password* yang dimasukan salah, maka sistem akan kembali lagi ke *form*

*login* untuk meminta *user* memasukan *password* yang benar. Apabila *username* dan *password* benar maka *user* akan masuk ke dalam sistem.

#### 4.2.3.3.2 Activity Diagram dari Use Case Kriteria

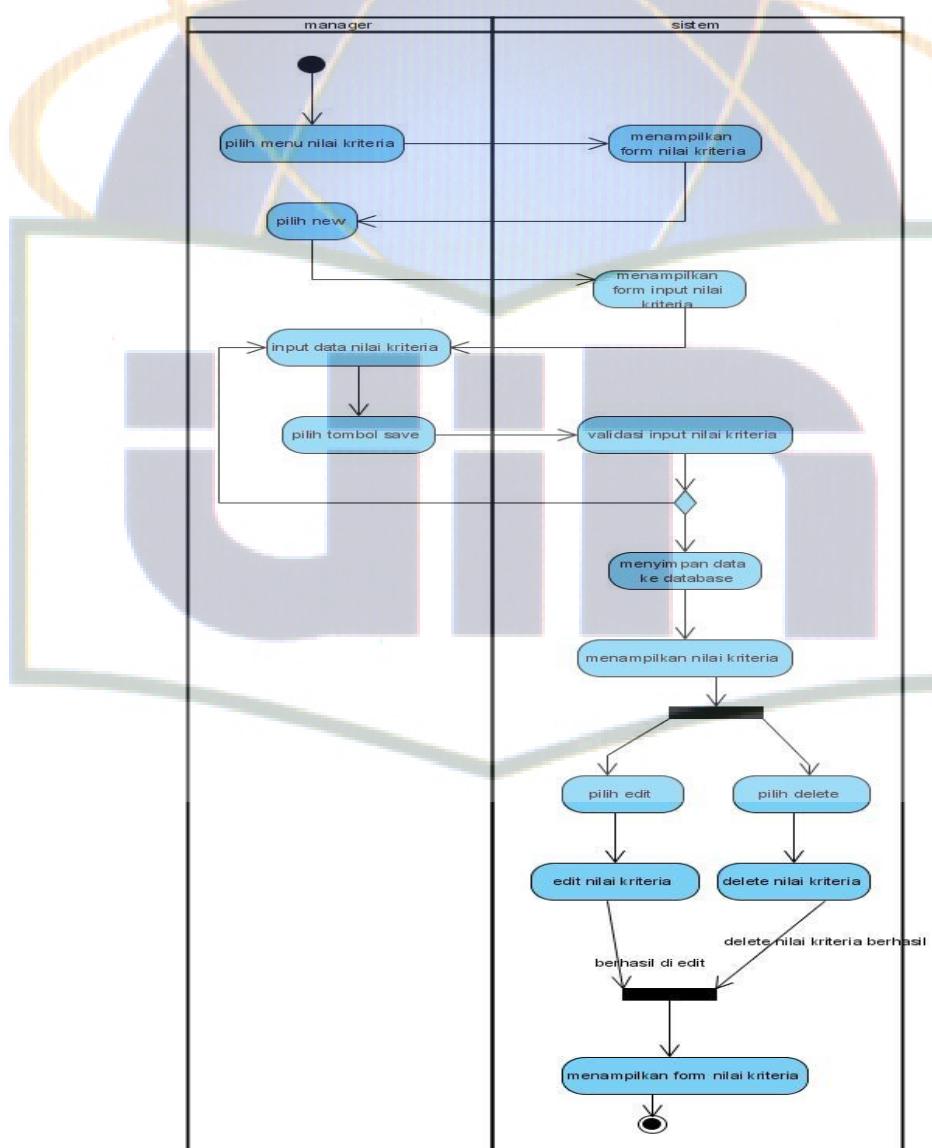


Gambar 4.12 – Activity Diagram dari Use Case Kriteria

Dalam *activity diagram* dari *use case* kriteria, *actor* pertama-tama melakukan login terlebih dahulu, membuka menu *utama* dan memilih kriteria pada menu. *Actor* insert dan edit atau menghapus kriteria yang ada, setelah selesai melakukan save agar menyimpan dalam database.

#### 4.2.3.3.3

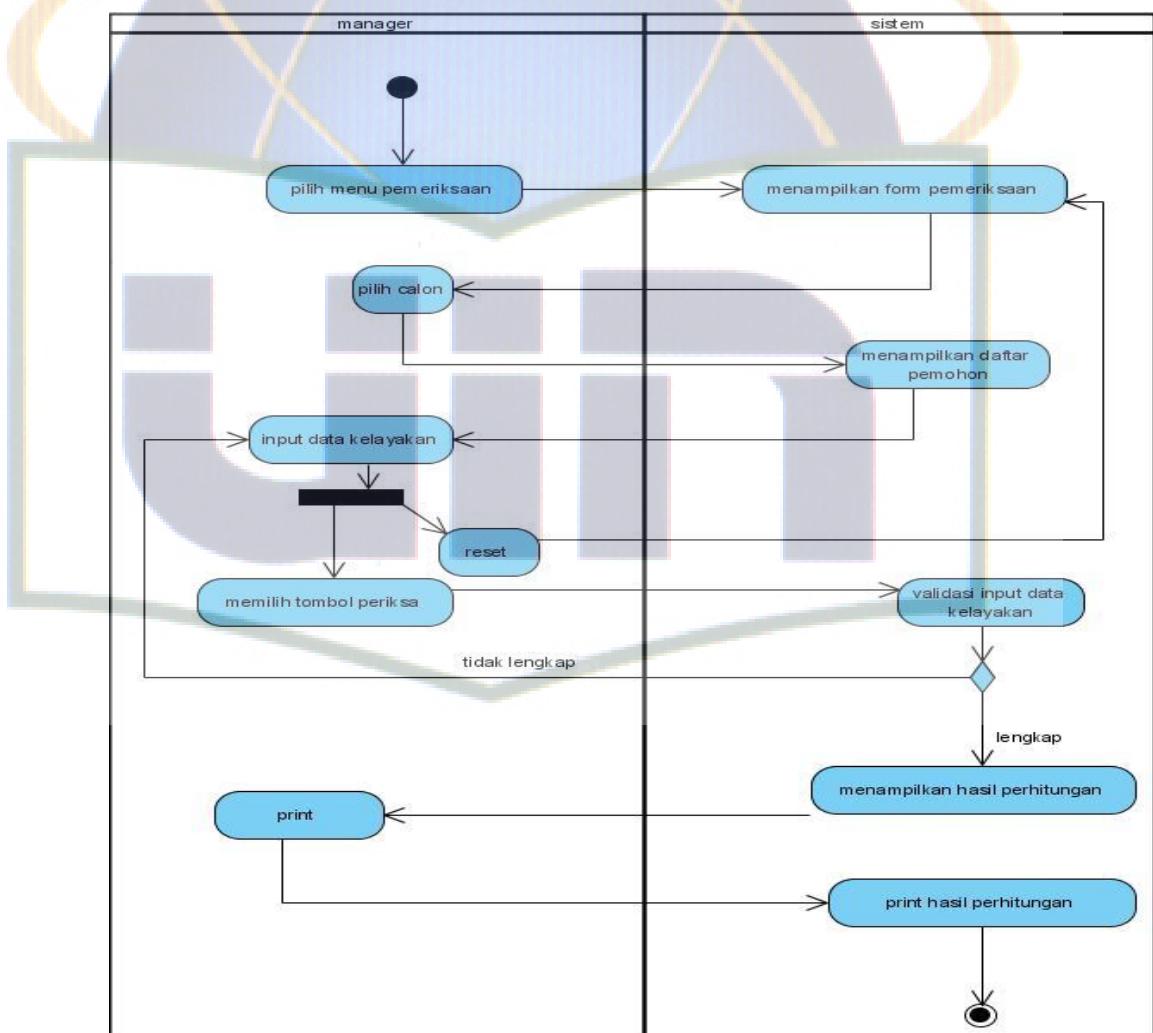
#### *Activity Diagram* dari *Use Case* Nilai Kriteria



Gambar 4.13 – *Activity Diagram* dari *Use Case* Nilai Kriteria

Dalam *activity diagram* dari *use case* nilai kriteria, sebelumnya *actor* pertama-tama login terlebih dahulu lalu membuka menu utama dan memilih nilai kriteria. *Actor* memilih kolom yang akan dinilai berdasarkan pada kolom yang kosong lalu di *save*. Jika nilai yang telah dimasukan ada yang kosong maka sistem akan menampilkan alert agar masukkan kembali nilai kriteria yang benar.

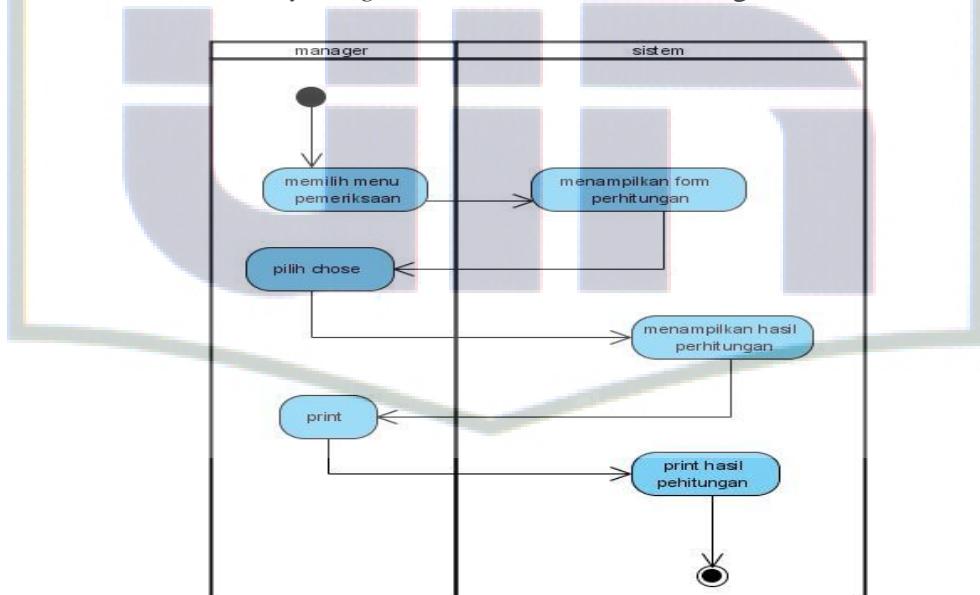
#### 4.2.3.3.4 Activity Diagram dari Use Case Pemeriksaan



Gambar 4.14 – Activity Diagram dari Use Case Pemeriksaan

Dalam *activity diagram* dari *use case* pemeriksaan, sebelumnya *manager* pertama-tama login terlebih dahulu, menampilkan menu utama lalu membuka menu *pemeriksaan* dan memilih nama pemohon yang akan dilakukan pemeriksaan perhitungan. *Manager* memilih jawaban yang sesuai dengan pemohon, dan menampilkan jawaban sebelum dihitung. *Manager* memilih hasil perhitungan kelayakan pembiayaan dengan menggunakan perhitungan *fuzzy* dan mendapatkan hasil *output* akhir berupa kelayakan pinjaman pembiayaan beserta informasi calon debitur dan dapat langsung keluar dari aplikasi.

#### 4.2.3.3.5 Activity Diagram dari Use Case Perhitungan

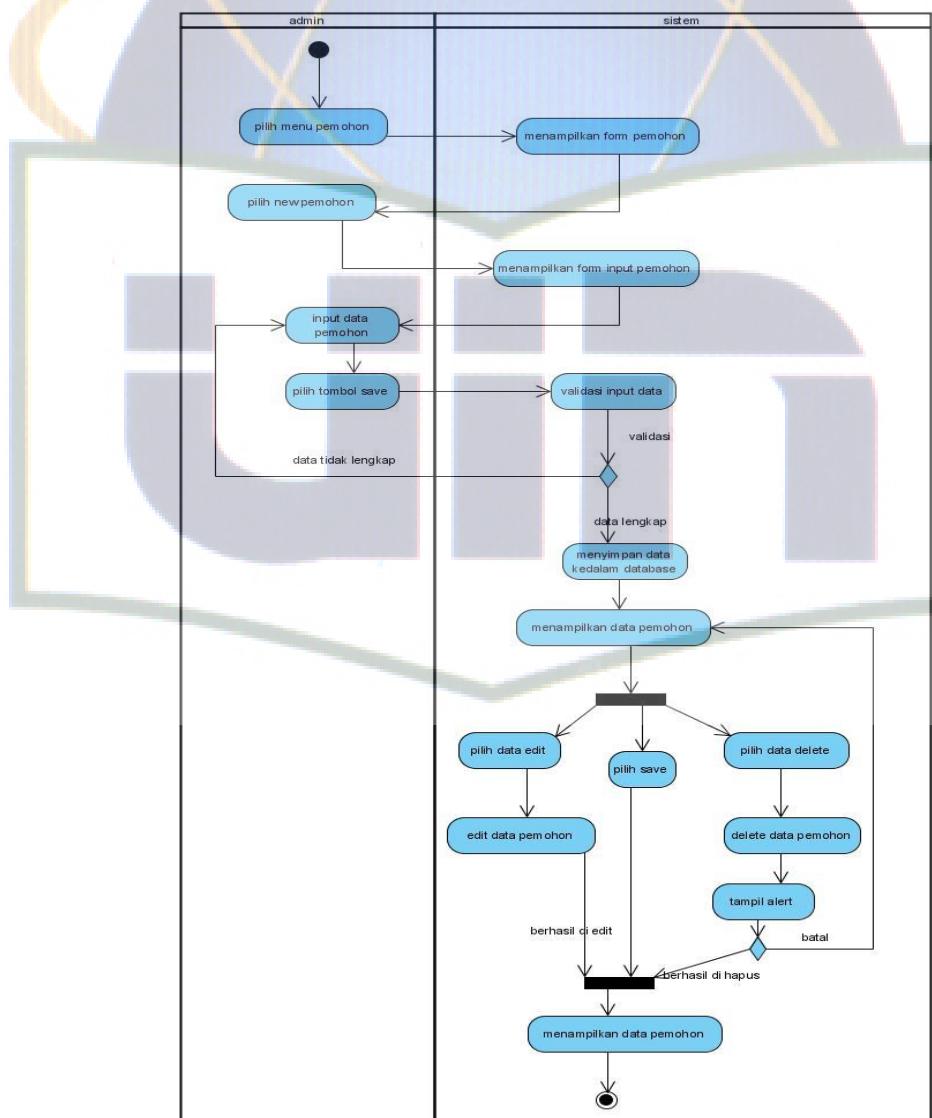


Gambar 4.15 – *Activity Diagram* dari *Use Case* Perhitungan

Dalam *activity diagram* dari *use case* perhitungan, sebelumnya *manager* pertama-tama login terlebih dahulu, menampilkan menu utama lalu membuka menu *pemeriksaan* dan memilih menu perhitungan dan akan

menampilkan form perhitungan, selanjutnya memilih nama pemohon yang akan lihat hasil perhitungan. *Manager* dapat mencetak dari hasil perhitungan yang didapatkan dari hasil *output* akhir berupa kelayakan pinjaman pemberdayaan beserta informasi calon debitur dan dapat langsung keluar dari aplikasi.

#### 4.2.3.3.6 Activity Diagram dari Use Case Informasi Calon Debitur

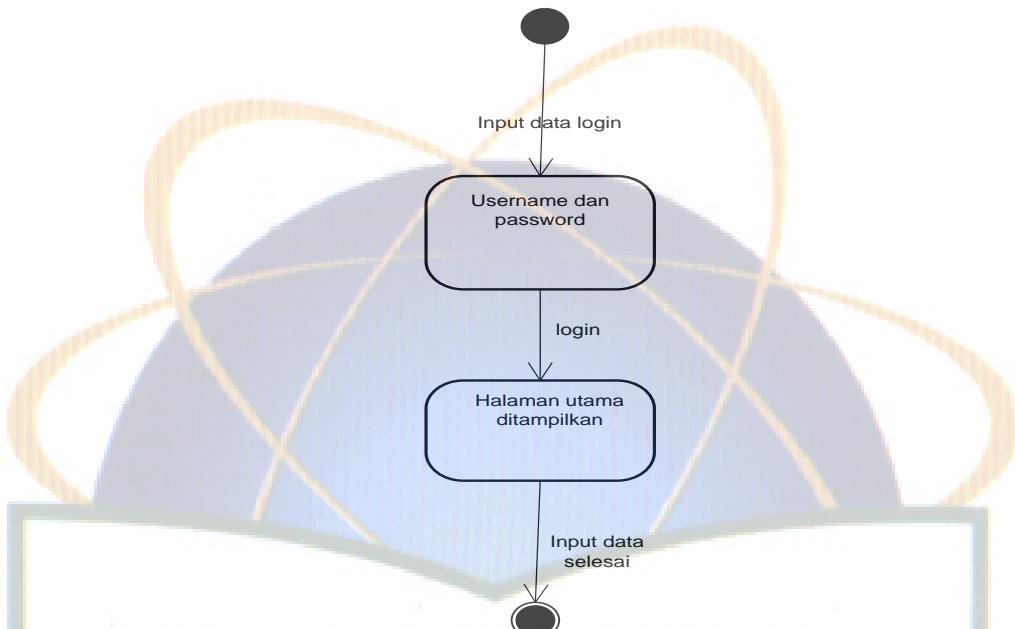


Gambar 4.16 – Activity Diagram dari Use Case Informasi Calon Debitur

Pada *activity diagram* di atas, *actor* pertama-tama membuka aplikasi tambahan pemohon yang disediakan yang berfungsi untuk menambahkan, mengubah, atau menghapus data-data calon debitur, baik yang akan dilakukan perhitungan kelayakan pembiayaan maupun seluruh nama nasabah yang terima dan tolak pembiayaan. Jika *actor* memilih insert, maka ia langsung memasukkan biodata calon debitur berupa nama, alamat, pekerjaan, telepon, usaha rencana penggunaan, nilai Jaminan, jangka waktu pembiayaan, nilai Jaminan, taksasi, administrasi. Jika *actor* memilih salah satu *field* calon debitur yang telah tersedia sebelumnya, maka ia dapat mengubah informasi pemohon tersebut. Jika *actor* ingin menghapus salah satu calon debitur, yakni dengan memilih salah seorang yang informasinya ingin dihapus, lalu memilih delete.

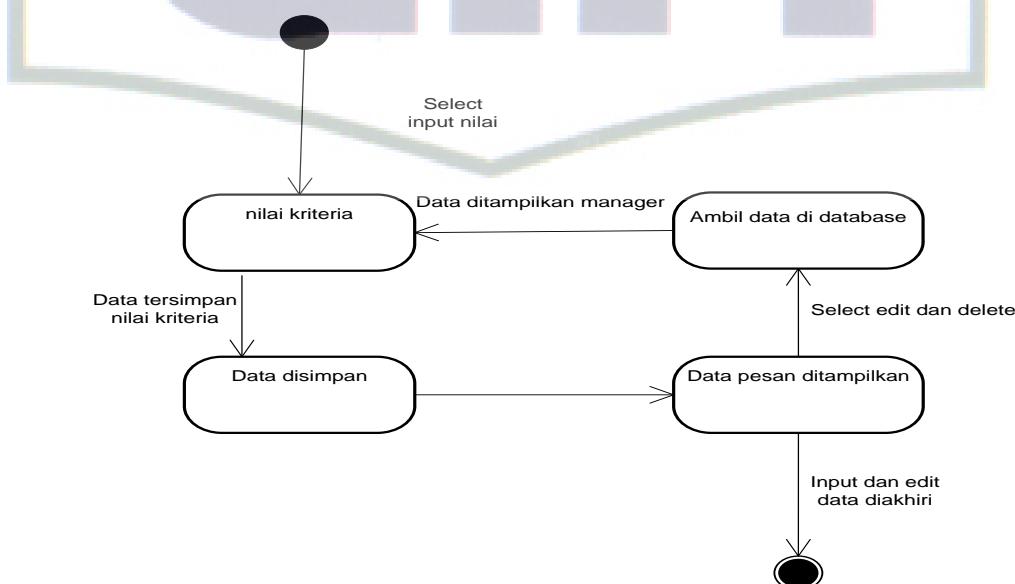
#### 4.2.3.4 Statechart Diagram

##### 1. Statechart Diagram dari Use Case Login



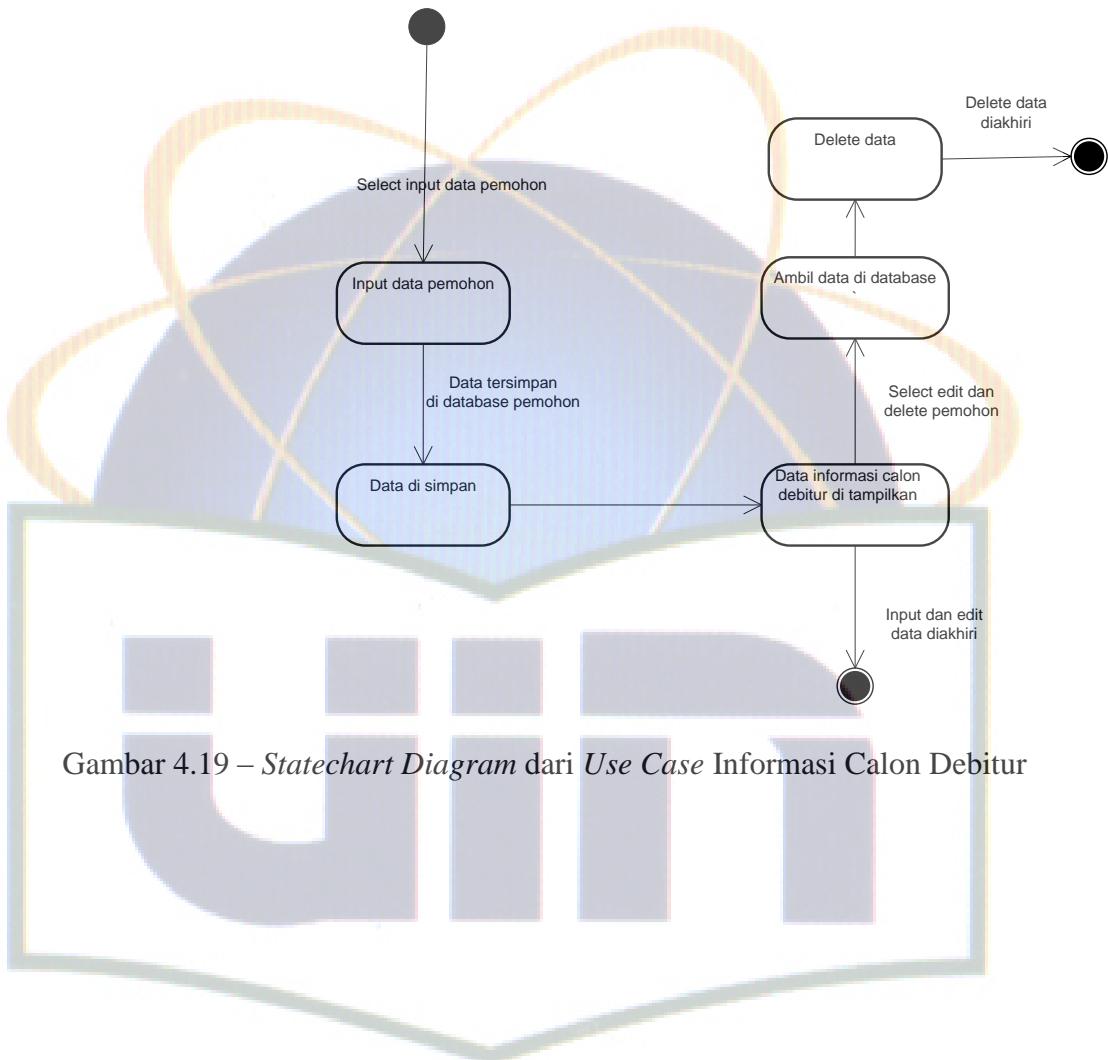
Gambar 4.17 Statechart Diagram Dari Use Case Login

##### 2. Statechart Diagram dari Use Case Nilai Kriteria



Gambar 4.18 – Statechart Diagram dari Use Case Nilai kriteria

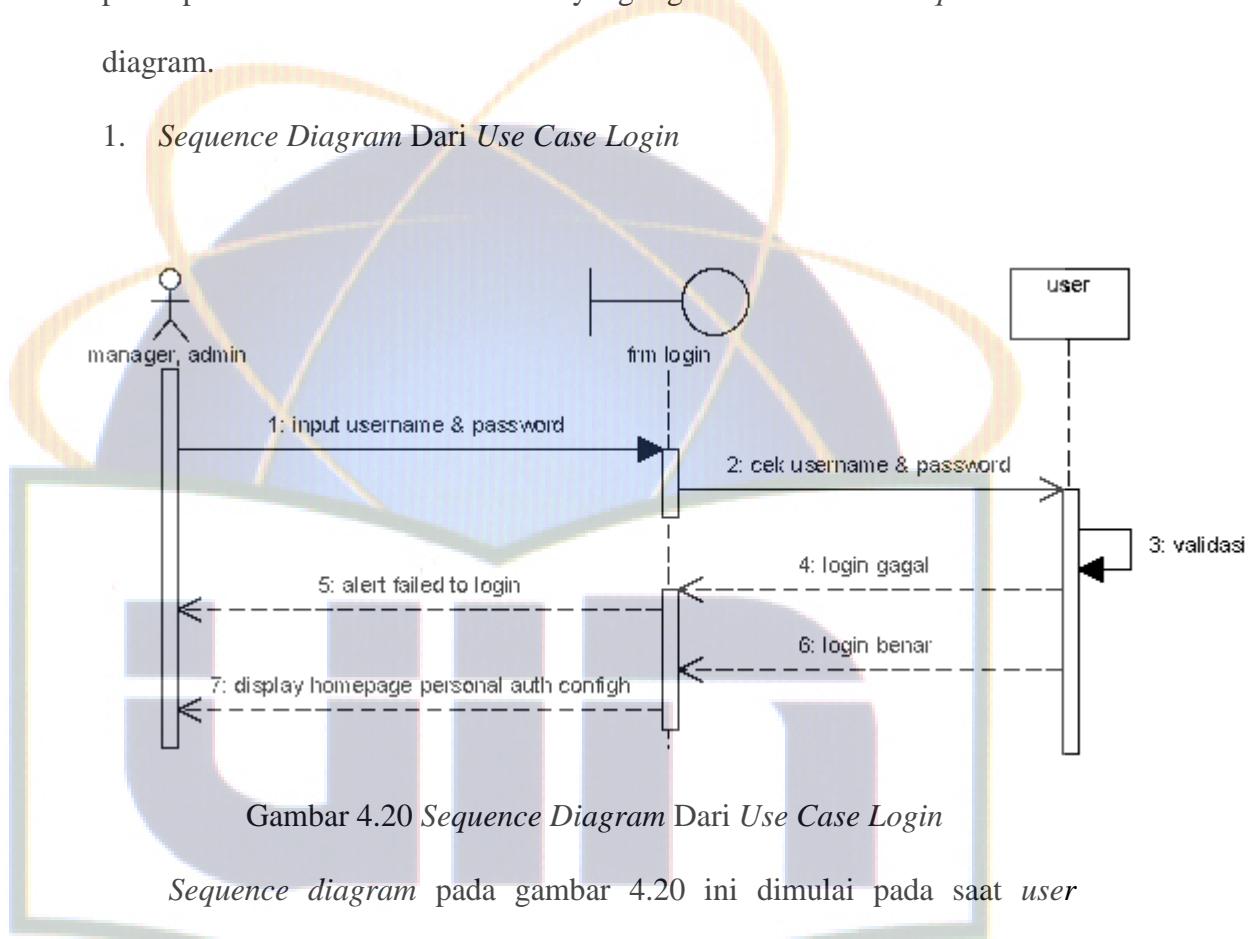
### 3. Statechart Diagram dari Use Case Mengisi Informasi Calon Debitur



#### 4.2.3.5 Sequence Diagram

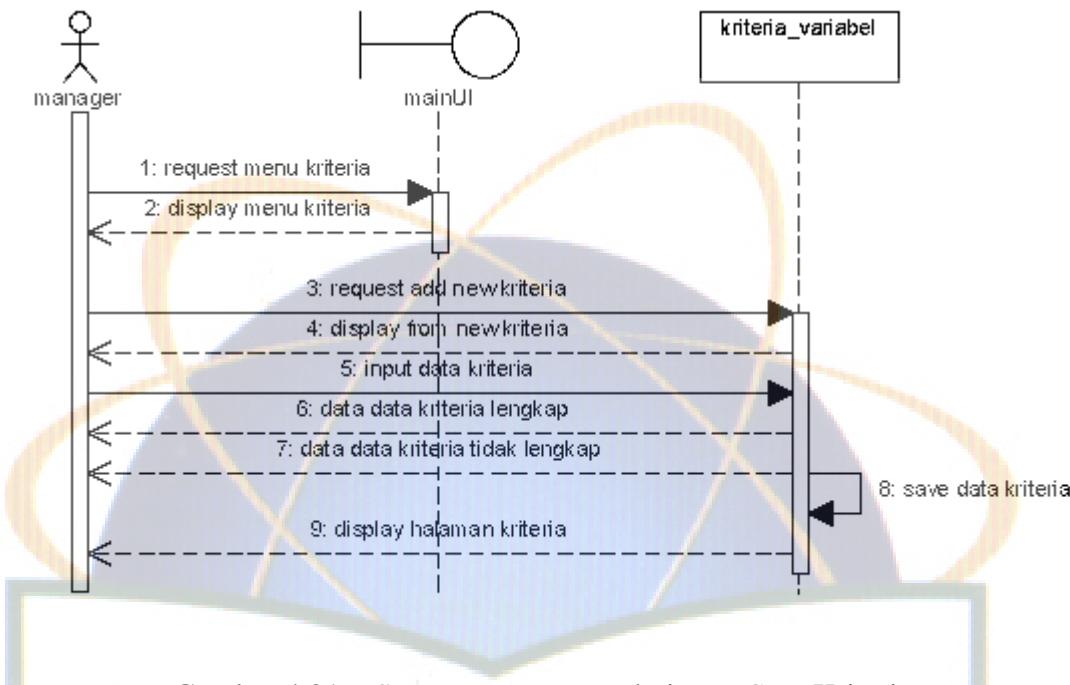
Pada sistem yang diajukan, terdapat interaksi antar *object* melalui pesan pada eksekusi sebuah usecase yang digambarkan dalam *sequence diagram*.

##### 1. Sequence Diagram Dari Use Case Login



Sequence diagram pada gambar 4.20 ini dimulai pada saat *user* (admin dan manager) akan menggunakan form *Login* untuk dapat masuk ke dalam sistem dan menggunakan objek *user* untuk dapat mengecek apakah *username* dan *password* yang dimasukan sesuai dengan yang ada pada *database* sistem. Jika *username* dan *password* yang diisi salah maka akan menampilkan *alert* dan *user* harus mengisinya kembali dengan *account* yang sesuai maka akan masuk ke dalam halaman utama *user* sesuai dengan pilihan *user* yang menggunakan sistem.

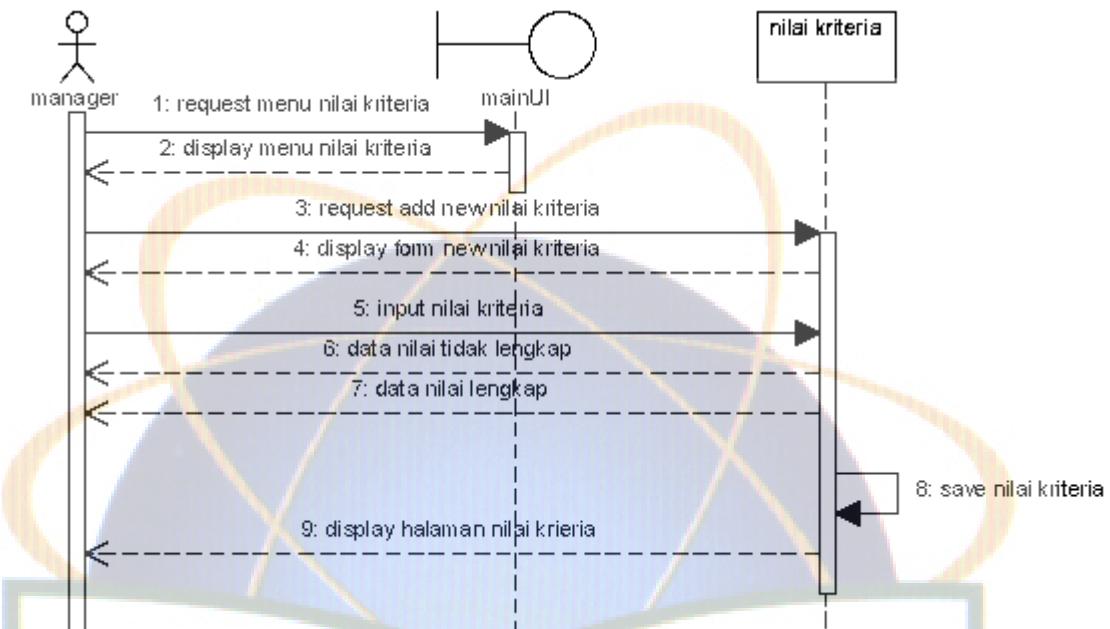
## 2. Sequence Diagram Dari Use Case Kriteria



Gambar 4.21 – Sequence Diagram dari Use Case Kriteria

*Sequence diagram* pada gambar 4.21 digunakan oleh Manager. Untuk manager dimulai pada saat manager menggunakan objek kriteria variabel untuk menulis data parameter, data akan diisi pada form kriteria. Jika data yang diisi tidak lengkap maka sistem akan menampilkan *alert* dan harus melengkapinya kembali. Setelah itu data kriteria parameter akan terbaca dan manager dapat melihat data kriteria parameter nya

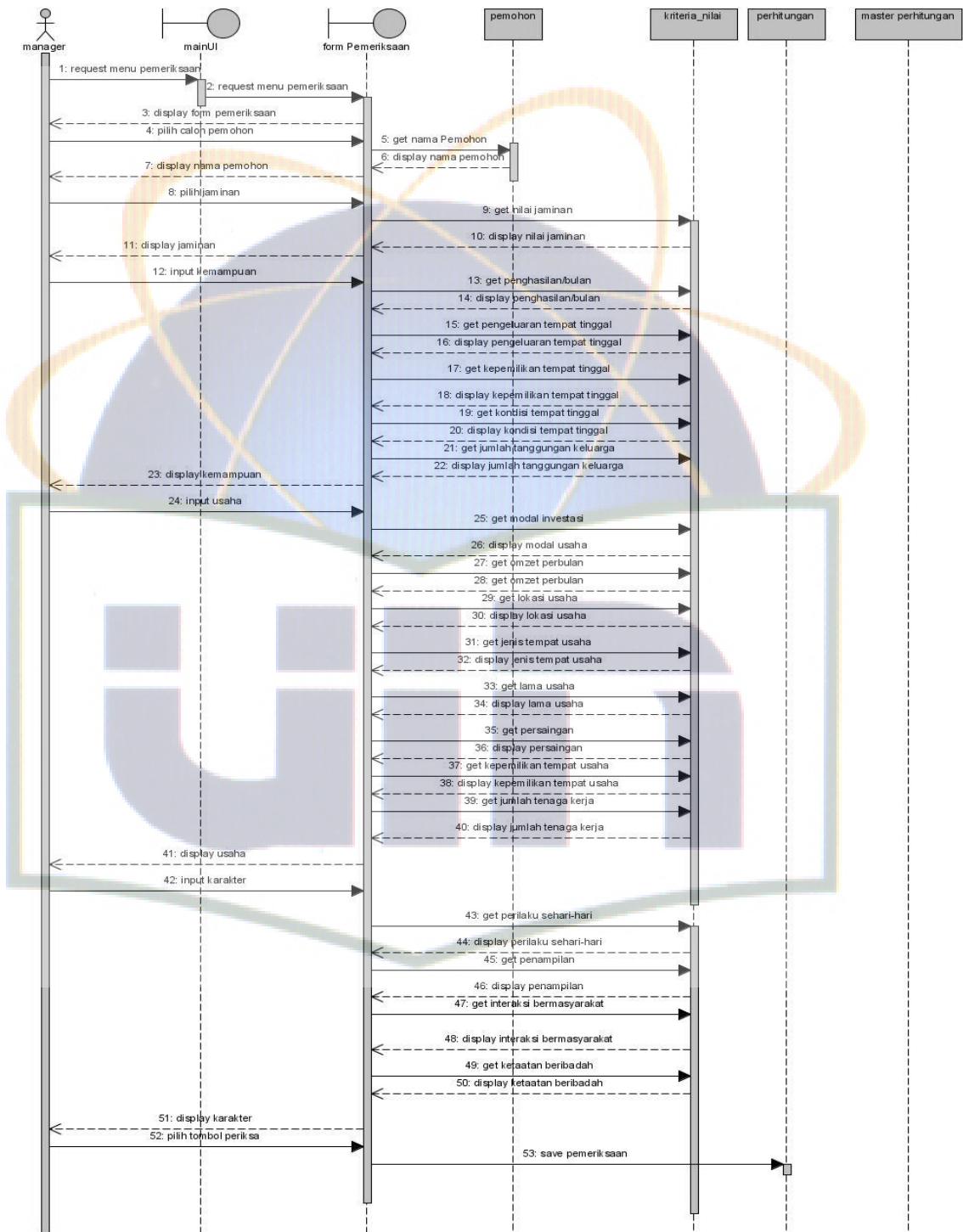
### 3. Sequence Diagram Dari Nilai Kriteria



Gambar 4.22- Sequence Diagram Use Case Nilai Kriteria

Gambar Sequence diagram pada gambar 4.22 digunakan oleh manager. Pada saat manager akan menulis data nilai kriteria, pada awalnya akan membaca objek kriteria terlebih dahulu untuk dapat mengetahui kriteria apa saja yang akan digunakan untuk nilai masing-masing kriteria tersebut. Lalu menggunakan objek nilai kriteria dan memasukkan nilai dan parameter sesuai dengan kriteria yang di pilih, jika data yang diisi tidak lengkap maka akan menampilkan *alert* dan manager harus melengkapinya. Setelah itu nilai parameter akan terbaca dan dapat melihat form nilai yang telah terisi semua.

#### 4. Sequence Diagram dari Use Case Pemeriksaan

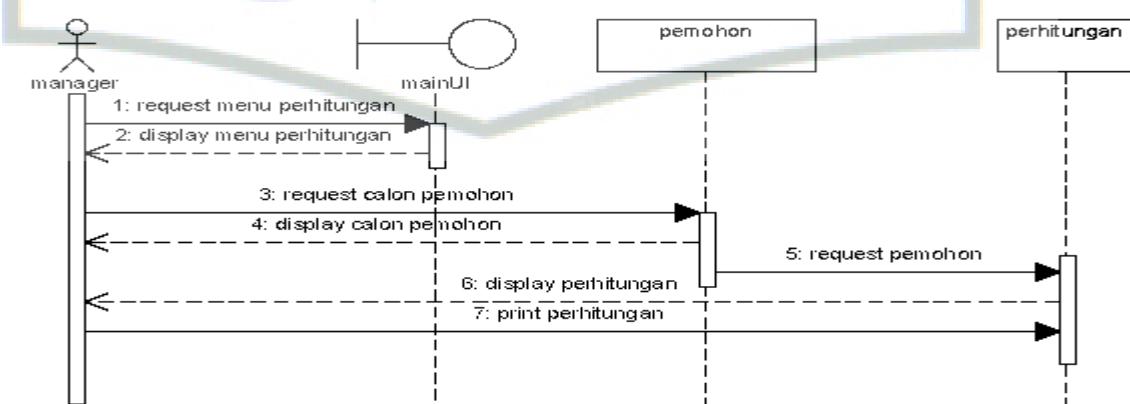


Gambar 4.23 Sequence Diagram dari Use Case Pemeriksaan

*Sequence diagram* pada gambar 4.23 digunakan oleh manager.

Dimulai pada saat manager akan perhitungan pembiayaan baru debitur maka sebelumnya manager harus mengetahui siapa nama debitur yang akan melakukan pengajuan pembiayaan dan kriteria apa yang cocok dalam parameter perhitungan. Sebelumnya manager harus memilih parameter pada objek kriteria apa saja yang cocok dalam data hasil survey yang didapat, lalu menggunakan objek perhitungan untuk mendapatkan nilai kelayakan, jika data belum terisi semua maka sistem akan menampilkan *alert* dan manager harus melengkapinya. Setelah itu perhitungan pembiayaan akan menghasilkan nilai dan manager dapat melihat data hasil perhitungan pembiayaan dan dapat mencetak data hasil perhitungan pembiayaan untuk laporan kelayakan pembiayaan debitur.

### 5. Sequence Diagram dari Use Case Perhitungan

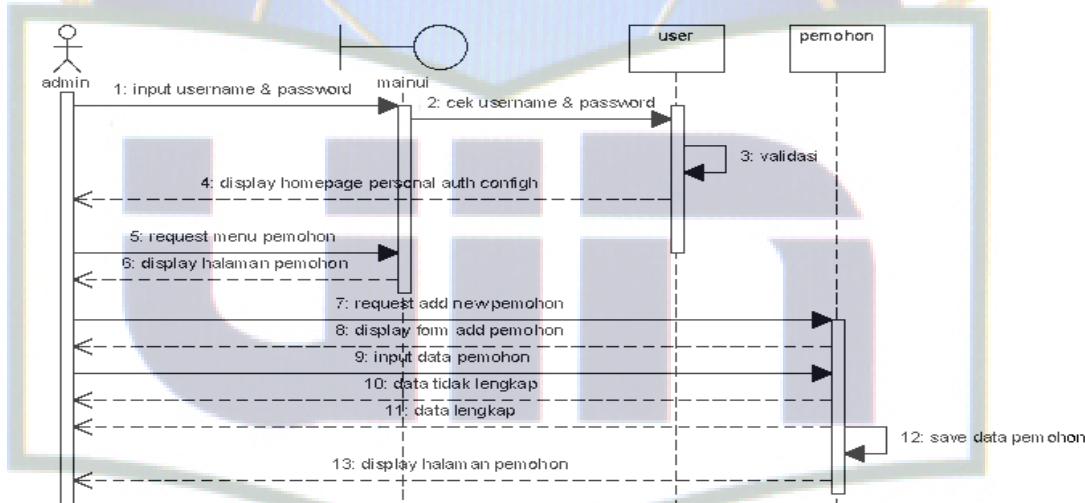


Gambar 4.24 *Sequence Diagram* dari Use Case Perhitungan

*Sequence diagram* dari *use case* perhitungan menunjukkan perilaku *actor* pada proses perhitungan fuzzy. *Message* membuka halaman menu

pemeriksaan dan membuka form hitung disampaikan kepada *object*. Didalam menu pemeriksaan terdapat objek kriteria nilai dan objek kriteria variabel. *Actor* kemudian mengirimkan *message* memilih nama calon debitur untuk menghitung nilai *fuzzy* kelayakan pembiayaan menentukan calon debitur yang akan mendapatkan pembiayaan. kemudian akan menjalankan perhitungan dan mengirimkan kembali *message* menampilkan *output* penilaian *fuzzy* kelayakan pembiayaan UKM informasi calon debitur kepada *actor*.

#### 6. Sequence Diagram dari Use Case Informasi Calon Debitur



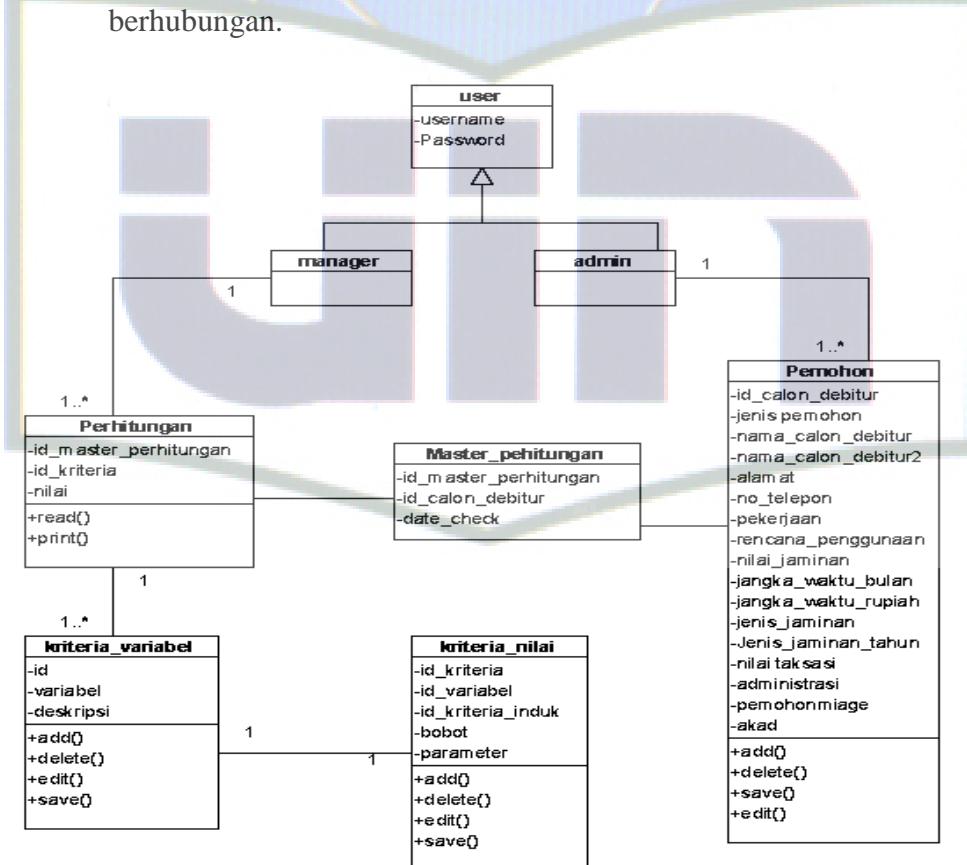
Gambar 4.25 – Sequence Diagram dari Use Case Informasi Calon Debitur

*Sequence diagram* pada gambar 4.25 digunakan oleh admin. Admin dimulai pada saat menggunakan objek pemohon untuk menulis data para calon debitur, data akan diisi pada form pemohon. Jika data yang diisi tidak lengkap maka sistem akan menampilkan *alert* dan harus melengkapinya kembali. Setelah itu data calon debitur akan terbaca dan admindapat melihat data pemohonnya

#### 4.2.3.6 Class Diagram

Class diagram menggambarkan struktur objek sistem, diagram ini menunjukkan kelas objek yang menyusun sistem dan juga hubungan antara kelas objek tersebut transisi (Whitten,2004 ).

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan dan interaksi diantara mereka. Diagram ini juga digunakan untuk mengorganisir objek-objek dari pemodelan use case dan mendokumentasikan hubungan diantara objek-objek tersebut. Bisa dibilang, class diagram adalah kumpulan dari beberapa objek yang saling berhubungan.



Gambar 4.26 Class Diagram

#### 4.2.4 Rancangan Database

##### 4.2.4.3 Nama Database

Nama *Database* yang akan dipakai dalam sistem penunjang keputusan untuk menilai kelayakan pembiayaan pada BPRS ini adalah *fuzzydb*. Nama ini dipakai karena ruang lingkup sistem adalah dengan menggunakan metode *fuzzy*.

##### 4.2.4.4 Identifikasi Database

Berikut adalah nama–nama tabel yang akan digunakan dalam *fuzzydb*. Tabel–tabel tersebut antara lain:

Tabel 4.16 Tabel Identifikasi *Database*

No	Nama Tabel
1	Login
2	Kriteria_nilai
3	Kriteria_variabel
4	Master_perhitungan
5	Pemohon
6	Perhitungan

Terdiri dari enam tabel berbeda yang akan dipakai dalam pembangunan sistem, yaitu; terdiri dari tabel login, tabel kriteria\_nilai, tabel kriteria\_variabel, tabel Pemohon, tabel master\_perhitungan, tabel Perhitungan, Masing – masing tabel akan dirancang dengan fungsi yang berbeda.

### 1. Tabel Login

File Name : Login

Primary Key : Id

File Type : Master\_Files

Tabel 4.17 Tabel Login

No	Field	Type	Size	Null	Keterangan
1	<i>Id</i>	Varchar	11	No	Id user
2	<i>Username</i>	Varchar	10	No	Username password
3	<i>Password</i>	Varchar	20	No	User password
4	<i>rule</i>	Varchar	20	No	<i>Rule fuzzy</i>

### 2. Tabel Kriteria\_nilai

File Name: Kriteria\_nilai

Primary Key: id\_kriteria

File Type: Master\_files

Tabel 4.18 Tabel Kriteria Nilai

No	Field	Type	Size	Null	Keterangan
1	<i>Id_kriteria</i>	Varchar	11	No	id kriteria
2	<i>Id_variabel</i>	Varchar	11	No	Id variabel
3	<i>Id_kriteria_induk</i>	Varchar	11	No	Id kriteria induk
4	Parameter	Varchar	100	No	Parameter yang digunakan
5	Bobot	Double	-	No	Nilai kriteria

### 3. Tabel Kriteria\_variabel

File Name: Kriteria\_variabel

Primary Key: id\_variabel

File Type: Master\_files

Tabel 4.19 Tabel Kriteria Variabel

No	Field	Type	Size	Null	Keterangan
1	Id	Varchar	11	No	id kriteria
2	Variabel	Varchar	50	No	Nama variabel
3	Deskripsi	Varchar	250	Yes	Analisa variabel

#### 4. Tabel Master\_perhitungan

File Name: master\_perhitungan

Primary Key: id\_master\_perhitungan

File Type: Master\_files

Tabel 4.20 Tabel Master Perhitungan

No	Field	Type	Size	Null	Keterangan
1	id_master_perhitungan	Varchar	11	No	id perhitungan
2	id_Calon_Debitur	Varchar	11	No	No pemohon/ Calon Debitur
3	Date_check	Datetime	-	Yes	Waktu

#### 5. Tabel Pemohon

File Name : pemohon

Primary Key : Id\_Calon\_Debitur

File Type : Master\_Files

Tabel 4.21 Tabel Pemohon

No	Field	Type	Size	Null	Keterangan
1	Id_Calon_Debitur	Varchar	11	No	Id Calon Debitur
2	Jenis_permohonan	Varchar	50	No	Jenis pemohonan
3	Nama_Calon_Debitur	Varchar	30	No	Nama pemohon
4	Alamat	Varchar	50	No	Alamat pemohon

5	No telepon	Varchar	50	Yes	No telepon pemohon
6	Pekerjaan	Varchar	50	No	Pekerjaan pemohon
7	Rencana_penggunaan	Varchar	50	No	Rencana penggunaan
8	Nilai_Jaminan	Varchar	15	No	Nilai jaminan
9	Jangka_waktu_bulan	Varchar	20	No	Jangka waktu perbulan
10	Jangka_waktu_rupiah	Varchar	15	Yes	Jangka waktu rupiah
11	Jenis_Jaminan	Varchar	20	No	Jenis jaminan
12	Jenis_Jaminan_thn	Varchar	10	Yes	Tahun Jenis jaminan
13	Nilai_taksasi	Varchar	15	No	Nilai taksiran
14	Adminstrasi	Varchar	20	No	Administrasi
15	Pemohonimage	Longblob	-	Yes	Foto pemohon
16	Akad	Varchar	60	No	Akad perjanjian

Tabel ini digunakan untuk menyimpan data pribadi dan Usaha pemohon pembiayaan.

## 6. Tabel Perhitungan

File Name : Perhitungan

Primary Key : id\_master\_perhitungan dan Id\_kriteria

File Type : Master\_Files

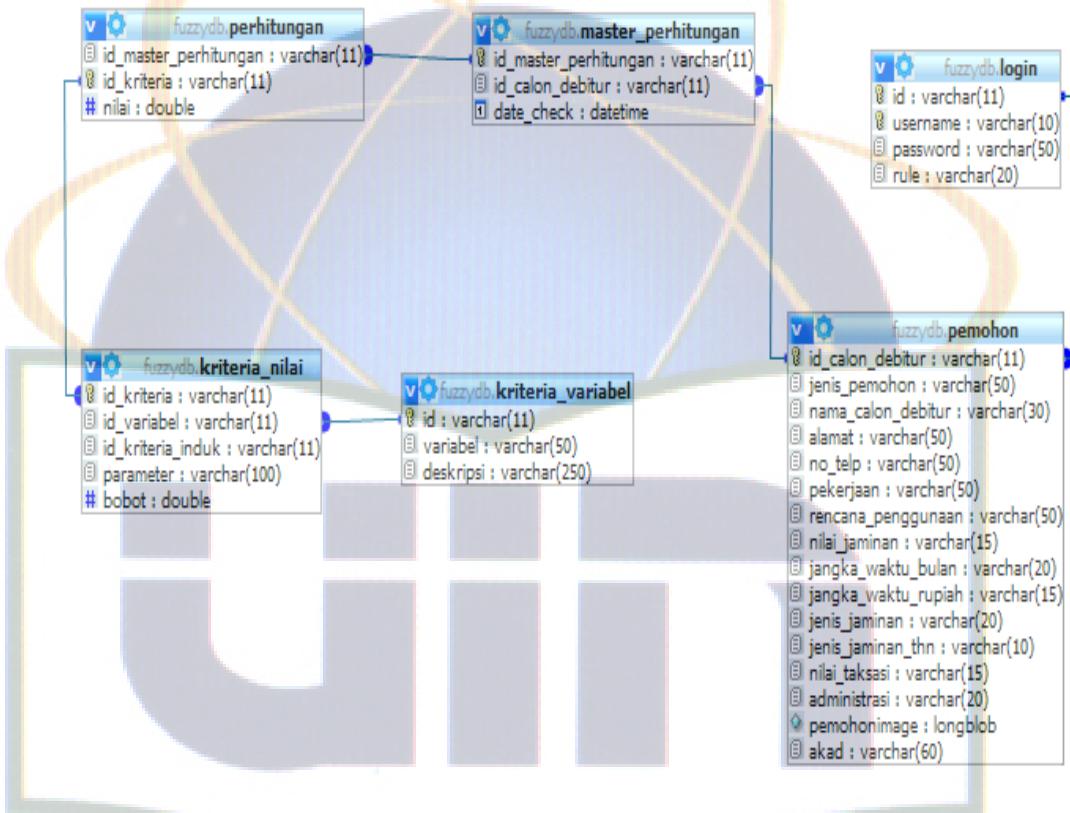
Tabel 4.22 Tabel Perhitungan

No	Field	Type	Size	Null	Keterangan
1	Id_master_perhitungan	Varchar	11	No	Id master perhitungan
2	Id_kriteria	Varchar	11	No	Id pemohon
3	Nilai	Double	-	No	Nilai dari variabel

#### 4.2.4.5 Physical Database Schema

Physical Database Schema atau Skema Database Fisik disajikan pada gambar di bawah ini:

Gambar 4.30 Physical Database Schema



Tahap yang dilakukan selanjutnya adalah perlunya identifikasi data dan hak akses apa yang diperlukan dan di lokasi mana. Analis sistem menemukan bahwa persyaratan logika tersebut perlu ditentukan dalam bentuk data-to- location-CRUD (*Create, Read, Update, Delete*) matrix yang merupakan suatu matriks yang digunakan untuk memetakan persyaratan data ke lokasi.

Tabel 4.23 Tabel *CRUD Database*

<b>Entity.Attribute</b>	<b>Location</b>	<b>Manager</b>	<b>Admin</b>
<b>Kriteria</b>		ALL	R
Id_kriteria		CRUD	R
Nama kriteria		CRUD	R
Nilai kriteria		CRUD	R
<b>Pemohon</b>		ALL	ALL
Id_Calon Debitur		R	CRUD
Jenis_permohonan		R	CRUD
Nama_Calon_Debitur		R	CRUD
Alamat		R	CRUD
No telepon		R	CRUD
Pekerjaan		R	CRUD
Rencana_penggunaan		R	CRUD
Nilai_Jaminan		R	CRUD
Jangka_waktu_bulan		R	CRUD
Jangka_waktu_rupiah		R	CRUD
Jenis_Jaminan		R	CRUD
Jenis_Jaminan_thn		R	CRUD
Nilai_taksasi		R	CRUD
Administrasi		R	CRUD
Akad		R	CRUD
<b>Pemeriksaan</b>		ALL	ALL
Id_periksa		CRUD	R
Id_Calon Debitur		CRUD	R
Nilai_Jaminan		CRUD	R
Modal_investasi		CRUD	R
Omzet		CRUD	R
Lokasi_Usaha		CRUD	R

Jenis_tempat_Usaha	<i>CRUD</i>	R
Lama_Usaha	<i>CRUD</i>	R
Persaingan	<i>CRUD</i>	R
Kepemilikan_tmpt_Usaha	<i>CRUD</i>	R
Jumlah_tenaga_kerja	<i>CRUD</i>	R
Penghasilan	<i>CRUD</i>	R
Pengeluaran	<i>CRUD</i>	R
Kepemilikan_tmpt_tinggal	<i>CRUD</i>	R
Kondisi_tmpt_tinggal	<i>CRUD</i>	R
Jumlah_tanggungan_klg	<i>CRUD</i>	R
Perilaku	<i>CRUD</i>	R
Penampilan	<i>CRUD</i>	R
Interaksi	<i>CRUD</i>	R
Ibadah	<i>CRUD</i>	R
<b>Hasil Perhitungan</b>	<b>ALL</b>	<b>ALL</b>
Id_hasil	R	R
Id_periksa	R	R
Tanggal	R	R
Rekomendasi	R	R
Jumlah_pinjaman	R	R

#### 4.2.5 Rancangan Antar Muka Pemakai

##### A. Rancangan Antar Muka Admin dan Manager

###### 1. Form Masuk

The diagram shows a login form with a light blue background. In the top-left corner is a placeholder box labeled 'IMAGE'. To its right is a large rectangular box containing the text: 'SISTEM PENUNJANG KEPUTUSAN KELAYAKAN PEMBIAYAAN USAHA KECIL MENEGAH' and 'BANK PEMBIAYAAN RAKYAT SYARIAH WAKALUMI'. Below this is a horizontal row with two buttons: 'ENTER' on the left and 'OUT' on the right. A large yellow circular arrow graphic is overlaid on the entire form.

Gambar 4.31 Rancangan *Form Masuk*

*Form* ini adalah tampilan utama sistem sebelum menuju ke *form* selanjutnya yaitu *form login*.

###### 2. Form Login

The diagram shows a login form with a light blue background. On the left is a placeholder box labeled 'IMAGE'. To its right is a large rectangular box containing the text: 'Isilah kolom dibawah ini dengan Benar'. Below this are two input fields: 'User Name' and 'Password', each with a corresponding text input box. At the bottom are two buttons: 'Login' on the left and 'Cancel' on the right. A large green diagonal arrow graphic is overlaid on the entire form, pointing from the top-left towards the bottom-right.

Gambar 4.32 Rancangan *Form Login*

Pada menu *login* ini merupakan halaman awal pembuka aplikasi Sistem Penunjang Keputusan Pembiayaan Kelayakan UKM. *User* diminta untuk memasukan *username* dan *password* dengan benar untuk menuju halaman utama sistem.

## B. Rancangan Antar Muka Admin

### 3. Form Utama

Keterangan.  
Petunjuk Menggunakan Aplikasi

- Pemohon  
Menu ini digunakan untuk input data pemohon pembiayaan, meliputi data pribadi, jumlah pembiayaan, jaminan, rencana penggunaan pembiayaan, serta kelengkapan persyaratan
- About  
Menu ini berisi informasi programer
- Exit  
Keluar dari Program

Gambar 4.33 Rancangan Form Utama Admin

Pada form menu utama ini merupakan halaman keterangan petunjuk masing-masing fungsi dari aplikasi Sistem Penunjang Keputusan Pembiayaan Kelayakan UKM.

### 4. Form Pemohon

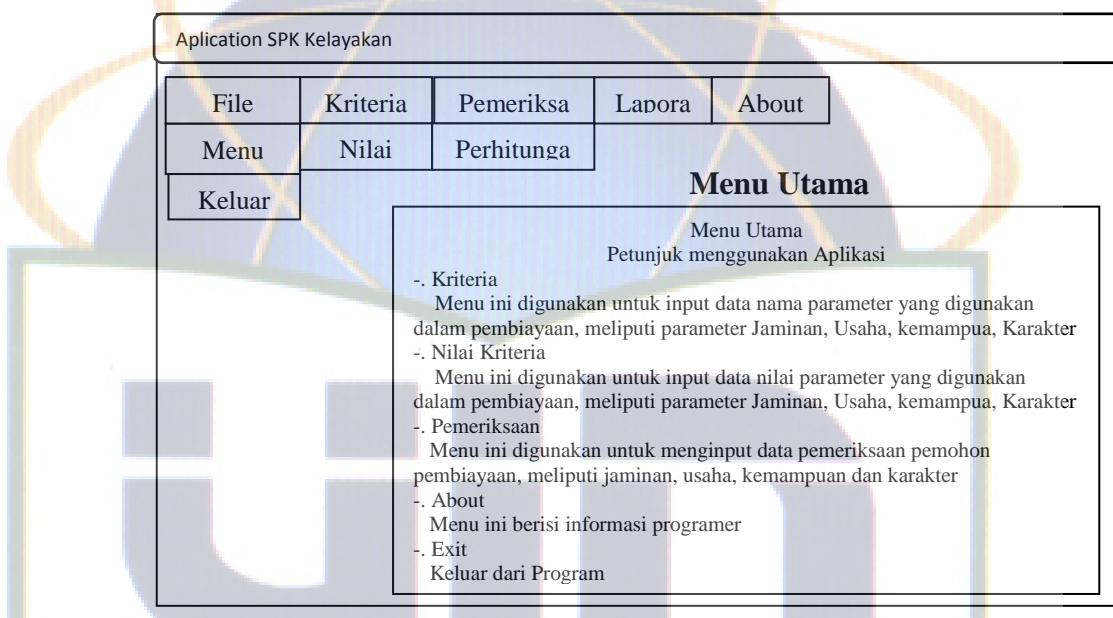
No		Rencana Pinjaman	
Jenis Pemohon	<input type="radio"/> Lama	Nilai Pinjaman	Rp
	<input type="radio"/> Baru	Jangka Waktu Pembiayaan	
Nama		Jenis barang jaminan	
Alamat Rumah		Nilai Taksasi jaminan	Rp
Pekerjaan /		Penyerahan Administrasi	
Akad		<input type="radio"/> Memenuhi	
		<input type="radio"/> Tidak memenuhi	
<input type="button" value="New"/> <input type="button" value="Hapus"/> <input type="button" value="Simpa"/> <input type="button" value="Refres"/>			

Gambar 4.33 Rancangan Form Pemohon

Form Pemohon digunakan untuk input data pemohon pembiayaan, yang meliputi data pribadi, jumlah pinjaman, Jaminan, rencana penggunaan pembiayaan, serta kelengkapan persyaratan.

### C. Rancangan Antar Muka Manager

#### 5. Form Utama Manager



Gambar 4.34 Rancangan *Interface Form Utama*

Pada form menu utama untuk manager ini merupakan halaman keterangan petunjuk masing-masing fungsi dari aplikasi Sistem Penunjang Keputusan Pembiayaan Kelayakan UKM.

## **6. Form Nilai Kriteria**

Application SPK Kelayakan

File	Kriteria	Pemeriksaan	Laporan	About
Menu utama	Nilai	Perhitungan		
Keluar				

Id.kriteria	Kri. variabel	Kriteria induk	Parameter	bobot

Variabel	<input type="text"/>
Deskripsi	<input type="text"/>

New	Hapus	Simpan	Refresh
-----	-------	--------	---------

Gambar 4.35 Rancangan manager in *Form Nilai Kriteria*

Pada form menu utama untuk ini merupakan halaman manage nilai kriteria untuk perhitungan, ketika ada perubahan dari data nilai parameternya.

## 7. Form Kriteria

Gambar 4.36 Rancangan *Form* Kriteria

Pada form menu utama untuk manager ini merupakan halaman manage nama kriteria yang ada untuk mengubah data, ketika ada perubahan dari data nama parameter yang telah ditetapkan perusahaan.

### 8. Form Pemeriksaan

File	Kriteria	Pemeriksaan	Laporan	About
No. Calon	<input type="text"/>	<input type="text"/>	<input type="button" value="Pilih Calon"/>	
<b>Jaminan</b>				
Nilai Jaminan	<input type="text"/>			
<b>Kemampuan</b>				
Penghasilan/bulan	<input type="text"/>	Kondisi Tempat Tinggal	<input type="text"/>	
Pengeluaran/bulan	<input type="text"/>	Jumlah Tanggungan Keluarga	<input type="text"/>	
Kepemilikan Tempat Tinggal	<input type="text"/>			
<b>Karakter</b>				
Modal Investasi	<input type="text"/>	Persaingan	<input type="text"/>	
Omzet/bulan	<input type="text"/>	Kepemilikan Tempat Usaha	<input type="text"/>	
Lokasi Usaha	<input type="text"/>	Jumlah Tenaga Kerja	<input type="text"/>	
Jenis Tempat Usaha	<input type="text"/>			
Lama usaha	<input type="text"/>			
<b>Usaha</b>				
Perilaku Sehari-Hari	<input type="text"/>	Interaksi bermasyarakat	<input type="text"/>	
Penampilan	<input type="text"/>	Ketaatan Beribadah	<input type="text"/>	
			<input type="button" value="Reset"/>	<input type="button" value="Periksa"/>

Gambar 4.37 Rancangan Form Pemeriksaan

Form pemeriksaan digunakan untuk input data pemeriksaan kondisi atau kriteria pemohon pembiayaan, yang meliputi jaminan, usaha, kemampuan, dan karakter.

## 9. Form Perhitungan

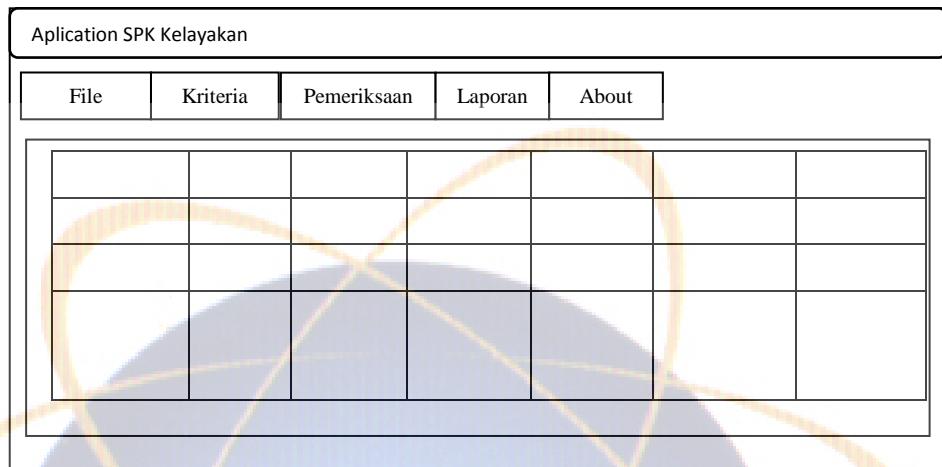
Form hasil digunakan untuk output hasil data pemeriksaan pemohon pemberian. Yang meliputi nilai jaminan, usaha, kemampuan, dan karakter pemohon pemberian, serta penilaian kelayakan hasil perhitungan dengan fuzzy. Dan jumlah pinjaman yang akan diberikan.

The diagram illustrates the layout of a calculation form (Form Perhitungan) for loan applications. The form is organized into several sections:

- Header:** Contains fields for "Calon" (Applicant) and "Date". A "Chose" button is located to the right.
- Jaminan (Collateral):** Includes fields for "Nilai" (Value) and "Jumlah Jaminan" (Amount of Collateral).
- Usaha (Business):** Includes fields for "Modal Investasi" (Investment Capital), "Omzet" (Turnover), "Lokasi usaha" (Business Location), "Lokasi Usaha" (Business Location), "Jenis tempat usaha" (Type of business location), "Lama usaha" (Business duration), "Persaingan" (Competition), "Jumlah tenaga kerja" (Number of workers), and "Jumlah Usaha" (Number of businesses).
- Kemampuan (Ability):** Includes fields for "Penghasilan/bulan" (Monthly Income), "Pengeluaran/bulan" (Monthly Expenses), "Kepemilikan Tempat Tinggal" (Ownership of residence), "Kondisi Tempat Tinggal" (Residence conditions), "Jumlah Tanggungan Keluarga" (Number of dependents), and "Jumlah Kemampuan" (Number of abilities).
- Karakter (Character):** Includes fields for "Perilaku Sehari-hari" (Daily behavior), "Penampilan" (Appearance), "Interaksi Bermasyarakat" (Community interaction), "Ketaatan Beribadah" (Religious obedience), and "Jumlah Karakter" (Number of characters).
- Results/Recommendations:** Includes fields for "Hasil Penilaian" (Evaluation results), "Rekomendasi Penilaian" (Evaluation recommendations), and "Jumlah Pinjaman Yang Akan Dipinjamkan" (Amount of loan to be borrowed).
- Buttons:** A "Print" button is located at the bottom right of the form area.

Gambar 4.38 Rancangan Form Perhitungan

## 10. Form Laporan



The diagram shows a user interface for a report form. At the top, there is a header bar with the text "Application SPK Kelayakan". Below the header is a navigation menu with five items: "File", "Kriteria", "Pemeriksaan", "Laporan", and "About". The main area contains a large grid structure, which appears to be a table or a matrix used for data analysis or reporting.

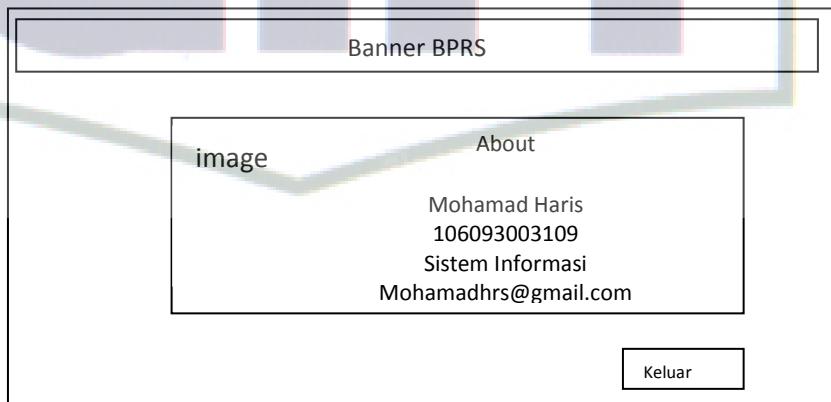
Gambar 4.39 Rancangan *Form Laporan*

Form ini berisikan seluruh nama peserta pembiayaan dan nilai yang telah dihasilkan dalam perhitungan dengan dilengkapi tanggal saat perhitungan.

## 11. Form About

Form ini berisikan informasi tentang pembuat program ini.

Rancangan form about sebagai berikut:



The diagram illustrates the layout of the "Form About". It features a "Banner BPRS" at the top. Below the banner is a section containing an "image" placeholder. To the right of the image is the word "About" followed by the author's details: "Mohamad Haris", "106093003109", "Sistem Informasi", and "Mohamadhrs@gmail.com". At the bottom right of the form is a button labeled "Keluar".

Gambar 4.40 Rancangan *Form About*

## 12. Tampilan Print

KESIMPULAN PEMERIKSAAN PERMOHONAN PEMBIAYAAN	
No.	: NSP001
Nama Calon Debitur	: Surya
Alamat	: Kwitang
Nila	: 8.821428571428571
Keputusan Penilaian : LAYAK DENGAN RESIKO RENDAH UNTUK DIPINJAMKAN	
Tangerang, 13/05/2011	
Manager	
(.....)	

Gambar 4.41 Rancangan Print Out Hasil

Form print out ini sebagai hasil dokumentasi perhitungan dari sistem ini yang memiliki nilai untuk mengambil keputusan



## 4.3 Fase Implementasi

### 4.3.1 Konstruksi Sistem Penunjang Keputusan

Dalam fase konstruksi, bahasa pemrograman yang digunakan adalah Java dengan menggunakan database MySQL serta penalaran *Fuzzy* metode *Tsukamoto*.

Instalasi yang dilakukan pada fase ini bertumpu pada sebuah jaringan komputer intranet dengan actor Manager dan Admin pemberian. Dalam proses instalasi, peranti lunak yang digunakan dan dilakukan penginstalan adalah XAMPP Ver. 1.6.8 dan JDK 6u12.

XAMPP Ver. 1.6.8 digunakan untuk mengontrol database MySQL pada sistem, sedangkan JDK versi 6u12 untuk menjalankan sistem yang menggunakan bahasa pemrograman Java.

Selain itu, juga menggunakan Netbeans dan Notepad++ sebagai *software editor*.

### 4.3.2 Pengujian Sistem Penunjang Keputusan

Kegiatan pada tahap implementasi ini meliputi pemrograman sistem dan pengujian sistem yang membahas sarana-sarana pendukung yang diperlukan agar sistem penilaian kelayakan ini dapat berjalan dengan baik dan modul-modul fungsi program.

Tabel 4.24 Tabel Pengujian Sistem Penilaian dengan Teknik Skoring dan Logika *Fuzzy*

<b>Kombinasi</b>	<b>Input</b>				<b>Output (Teknik Scoring)</b>		<b>Output (Logika Fuzzy)</b>	
	<b>Jmn</b>	<b>Ush</b>	<b>Kmp</b>	<b>Krktr</b>	<b>Hasil</b>	<b>Ket</b>	<b>Hasil</b>	<b>Ket</b>
Kombinasi 1	10	6.75	5.25	6.25	7,0625	Peninjauan kembali	8,8214287	Layak Resiko
								Rendah
Kombinasi 2	10	5	1.25	10	6,5625	Layak Resiko Tinggi	5,3125	Layak Resiko Tinggi
Kombinasi 3	10	6.5	7.25	10	8,4375	Pembiayaan Layak diberikan	8,791666	Layak Resiko Rendah
Kombinasi 4	5	10	2.75	1.25	4,75	Pembiayaan Ditolak	3,53125	Kurang / Tidak Layak
Kombinasi 5	0	8.2	2.5	2.5	3,3125	Pembiayaan Ditolak	0,625	Sangat Tidak Layak

Dari semua uji coba diatas, dapat disimpulkan bahwa sistem sudah sesuai dengan spesifikasi kebutuhan pengguna. Penilaian dengan logika fuzzy dapat menghasilkan nilai yang lebih fleksibel dibandingkan penilaian dengan teknik scoring yang cenderung kaku. Hal ini terlihat pada *output* yang dihasilkan, dengan logika fuzzy klasifikasi kelayakan disediakan dalam 5 kategori kelayakan, walaupun skor yang dihasilkan berbeda, namun perbedaan tersebut tidak begitu signifikan dengan nilai yang dihasilkan dengan logika fuzzy.

Pada tahap pengujian Sistem Penunjang Keputusan, pengujian dilakukan dengan cara Blackbox Testing. Cara pengujian dilakukan dengan menjalankan aplikasi Sistem Penunjang Keputusan dan melakukan input data serta melihat pengujian model dilakukan oleh pakar Bank Pembiayaan Rakyat Syariah WAKALUMI. Pengujian ini diperlukan untuk mengetahui apakah sistem telah sesuai dengan spesifikasi kebutuhan pengguna. Pengujian dilakukan sebanyak lima kali dengan berbagai variasi masukan dan keluaran. namun kombinasi yang dilakukan hanya untuk

pemohon dengan kualifikasi tingkat kemampuan menengah kebawah. Hal ini disebabkan karena pakar berasumsi bahwa untuk pemohon dengan tingkat Kemampuan menengah ke atas akan memiliki jawaban Sangat Bagus untuk sebagian kriteria sehingga menjamin bahwa pemohon tersebut sangat layak atau layak dengan resiko rendah untuk memperoleh pemberian. Hasil pengujian blackbox testing disajikan dalam tabel di bawah ini:

Tabel 4.25 Pengujian dengan Blackbox Testing

No	Rancangan Proses	Hasil yang diharapkan	Hasil	Keterangan
1	Masukkan Password dan Username lalu klik login	Akan tampil halaman menu utama berisikan petunjuk aplikasi	Sesuai	
2	Salah masukkan Password dan Username lalu klik login	Akan tampil dihalama muka login gagal	Sesuai	
3	Select Pemohon	Akan tampil menu Pemohon pada halaman muka	Sesuai	
4	Select Menu Utama	Akan menampilkan Petunjuk Masing-masing Keterangan pada menu yang tersedia	Sesuai	
5	Klik New pada Pemohon	Isian data pemohon kedalam sistem dan tampil kehadapan muka	Sesuai	
6	Klik Delete pada Pemohon	Isian data pemohon kedalam sistem dan tampil kehadapan muka	Sesuai	
7	Klik Refresh pada Pemohon	Data Pemohon yang telah masuk dapat diperbarui.	Sesuai	
8	Klik Save pada Pemohon	Data Pemohon yang tersimpan dapat diubah dan tampil pada halaman muka	Sesuai	
9	Input Nilai Kriteria	Akan Tampil Form isian Nilai Kriteria pada halaman muka.	Sesuai	
10	Klik New pada Nilai	Isian Nilai Kriteria kedalam sistem	Sesuai	

	Kriteria	dan tampil kehadapan muka		
11	Klik Delete pada Kriteria	Isian Nilai Kriteria kedalam sistem dan tampil kehadapan muka	Sesuai	
12	Klik Refresh pada Kriteria	Data Nilai Kriteria yang telah masuk dapat diperbarui.	Sesuai	
13	Klik Save pada Kriteria	Data Nilai Kriteria yang tersimpan dapat diubah nilai dan tampil pada halaman muka	Sesuai	
14	Select Kriteria	Akan Tampil Form isian nama Kriteria dan keterangan analisa pada halaman muka.	Sesuai	
15	Klik New pada Kriteria	Isian Nama Kriteria kedalam sistem dan tampil kehadapan muka	Sesuai	
16	Klik Delete pada Kriteria	Isian Nama Kriteria kedalam sistem dan tampil kehadapan muka	Sesuai	
17	Klik Refresh pada Kriteria	Data Nama Kriteria yang telah masuk dapat diperbarui.	Sesuai	
18	Klik Save pada Kriteria	Data Nama Kriteria yang tersimpan dapat diubah nilai dan tampil pada halaman muka	Sesuai	
19	Select Pemeriksaan	Akan menampilkan Form isian nama Pemohon dan isian Parameter untuk melakukan perhitungan pada halaman muka.	Sesuai	
20	Klik Pilih Calon pada Pemeriksaan	Akan menampilkan nama-nama pemohon untuk dilakukan perhitungan	Sesuai	
21	Klik Reset pada Pemeriksaan	Tampilan dikosongkan dari halaman muka	Sesuai	
22	Klik Periksa/Hitung pada Pemeriksaan	Akan menampilkan form nilai dari yang telah disikan akan mengeluarkan hasil perhitungan	Sesuai	
23	Select Perhitungan	Akan menampilkan Form nilai perhitungan halaman muka.	Sesuai	
24	Klik Print pada halaman perhitungan	Akan menampilkan form hasil perhitungan untuk dicetak	Sesuai	
25	Select Laporan	Akan menampilkan seluruh pemohon dan nilai yang dihasilkan	Sesuai	
26	Select About	Akan menampilkan pembuat aplikasi	Sesuai	

### 4.3.3 Sarana Pendukung Sistem

Kebutuhan sistem ini dapat dibangun dengan menggunakan pada komputer yang mempunyai spesifikasi sebagai berikut:

a. Perangkat Keras

Sebuah komputer untuk yang digunakan bagian Marketing BPRS

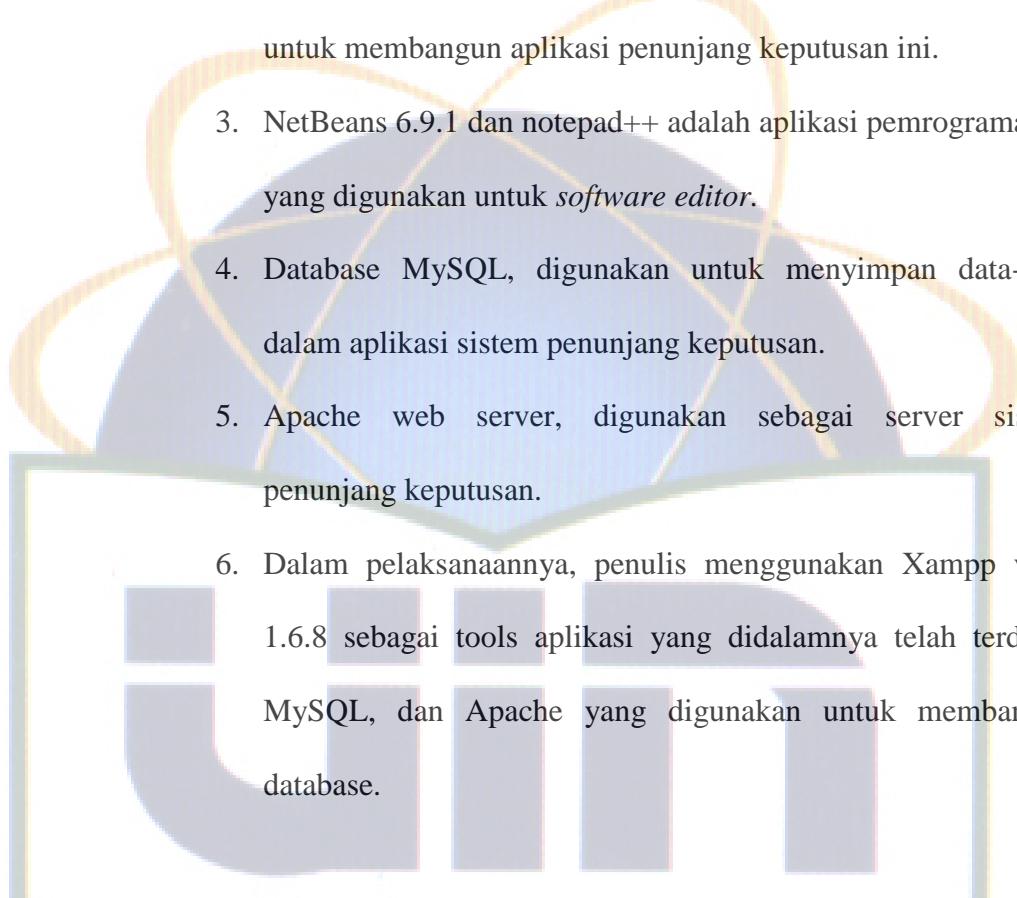
WAKALUMI untuk menjalankan aplikasi. Spesifikasi minimum komputer yang dibutuhkan adalah:

1. Pentium 4 Processor 3.06 GHz
2. Memory minimal 512 MB
3. Keyboard dan Mouse
4. Monitor 15 "
5. Hardisk 20 GB

b. Perangkat Lunak

Sistem penunjang keputusan ini tidak dapat dibangun tanpa disertai perangkat lunak. Dengan adanya perangkat lunak komputer akan bekerja memproses data dan menghasilkan informasi. Perangkat lunak ini terdiri dari perangkat lunak sistem operasi (*Operating System*) dan perangkat lunak aplikasi (*Application Software*).

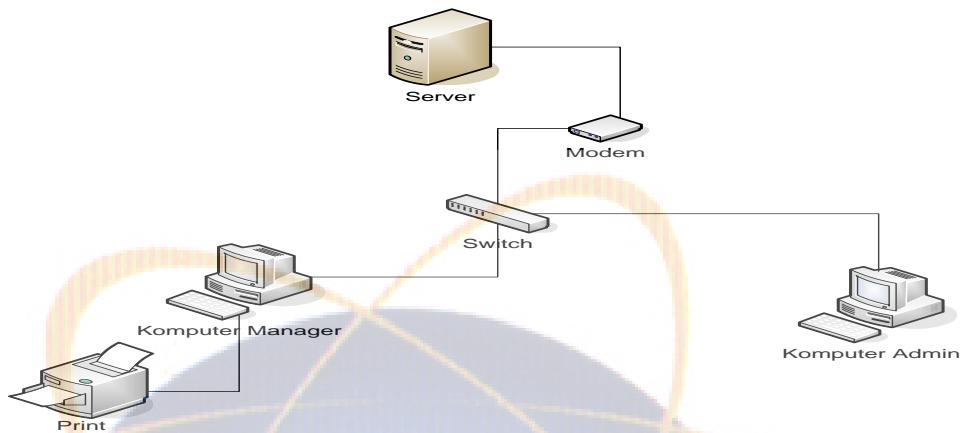
Sistem penunjang keputusan ini dapat dijalankan di semua *operation system* yang telah mensupport aplikasi Java. Perangkat lunak yang dibutuhkan adalah sebagai berikut:

- 
1. Operation sistem yang digunakan adalah Windows XP/Vista/Seven.
  2. Bahasa pemrograman yang dibunakan adalah Java, digunakan untuk membangun aplikasi penunjang keputusan ini.
  3. NetBeans 6.9.1 dan notepad++ adalah aplikasi pemrograman yang digunakan untuk *software editor*.
  4. Database MySQL, digunakan untuk menyimpan data-data dalam aplikasi sistem penunjang keputusan.
  5. Apache web server, digunakan sebagai server sistem penunjang keputusan.
  6. Dalam pelaksanaannya, penulis menggunakan Xampp versi 1.6.8 sebagai tools aplikasi yang didalamnya telah terdapat MySQL, dan Apache yang digunakan untuk membangun database.

### c. Jaringan *intranet* Sistem Penunjang Keputusan Pembiayaan UKM

Server yang digunakan di kontrol oleh staff bagian IT diruang Teknis BPRS Wakalumi, dimana seluruh kegiatan teknis dan Teknologi komputer berada diruang ini.

PC milik *manager* berada di ruangan marketing dan PC admin pun berada di ruangan marketing. Dimana seluruh komputer terhubung melalui jaringan intranet dalam perusahaan.



Gambar 4.42 Jaringan *intranet* Sistem Penunjang Keputusan Pembiayaan UKM

#### 4.3.4 Kelebihan

Sistem ini mampu melakukan penilaian kelayakan pembiayaan dengan klasifikasi penilaian yang disediakan dalam lima pilihan yaitu sangat layak, layak resiko rendah, layak resiko tinggi, kurang layak, sangat tidak layak. Sehingga *User* lebih dapat mengekspresikan penilaian mereka dibandingkan dengan tiga penilaian saja misalnya sangat layak, cukup layak, dan tidak layak.

Sistem ini ditampilkan dalam suatu *user interface* yang didesign sedemikian rupa sehingga bersifat *user friendly*.

#### 4.3.5 Kekurangan

Dalam sistem penunjang keputusan pembiayaan dengan Logika Fuzzy dengan metode Tsukamoto lebih cenderung tidak transparan dalam melakukan perhitungan program karena tidak ditampilkan halaman dalam program ini.

## BAB V

### KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan skripsi beserta saran untuk penelitian lebih lanjut:

#### 5.1 Kesimpulan

Berdasarkan uraian dan pembahasan pada bab-bab sebelumnya, maka dapat disimpulkan bahwa:

1. Sistem Penunjang Keputusan Kelayakan Pembiayaan dengan logika *fuzzy* dapat digunakan sebagai alternatif dalam mengevaluasi kelayakan calon debitur untuk memperoleh pembiayaan akurat yang lebih baik .
2. Dengan logika *fuzzy* , sistem memberikan hasil yang lebih fleksibel dalam membantu menentukan calon debitur berdasarkan kriteria jaminan, usaha, kemampuan dan karakter. Sehingga memberikan kemudahan bagi calon debitur dengan tetap memperhitungkan kelayakannya.

## 5.2 Saran

Berdasarkan kesimpulan yang telah dipaparkan di atas. Untuk penelitian lebih lanjut, maka penulis menyarankan:

- Sistem dapat dikembangkan lebih lanjut dengan menyediakan lebih banyak lagi berbagai macam kriteria yang sekiranya lebih dibutuhkan dalam perusahaan yang membutuhkan penilaian kelayakan pembiayaan, karena setiap perusahaan memiliki ketentuan dan nilai tersendiri dari parameter perusahaan pembiayaannya, sehingga dapat digunakan oleh berbagai Perusahaan lainnya.

## DAFTAR PUSTAKA

- Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: Andi.
- Amalia, Lia. 2010. *Model Fuzzy Tahani untuk Pemodelan Sistem Pendukung Keputusan (SPK) Kasus: Rekomendasi Pembelian Handphone*. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010):1907-5022.
- Anoraga, Pandji, et. al. 2002. *Koperasi, Kewirausahaan Dan Usaha Kecil*. Jakarta: Rineka Cipta.
- Asnur, Daniel. 2009. *Penyusunan Decision Support System (DSS) Studi Kelayakan Ekonomi dan Finansial bagi UMKM*. Kepala Bidang Kerja sama dan Jaringan pada Deputi Bidang Pengkajian Sumber daya UKMK. Jurnal Volume 4.
- Cahyono, Setiyo. 2006. *Panduan Praktis Pemrograman Database Menggunakan MySQL dan Java*. Bandung: Informatika.
- Dhiana, Aziz Sukma. 2010. *Rancang Bangun Sistem Pakar Untuk Mendeteksi Gizi Buruk Pada Balita*. UIN Syarif Hidayatullah Jakarta.
- (DRPM UI). 2009. *Pedoman Monitoring Dan Evaluasi Dosen Inti Penelitian Universitas Indonesia*. Bidang Penelitian Dan Publikasi.
- Glass, Michael, et. al. 2004. *Beginining PHP, Apache, MySQL Web Development*. Indianapolis: Wiley Publishing.
- Gulo, W. 2002. *Metodologi Penelitian*. Jakarta: Grasindo.

- Herri, et, al. 2008. *Studi peningkatan peran Bank perkreditan rakyat (BPR) dalam usaha mikro kecil (UMK) di Sumatra Barat.* Center for Banking Research (CBR)-Andalas University.
- Hiemann, Wolfram, 2002. *Scoring a profitable tool for SME-lending and loan risk management.* Jakarta: Kantor Meneg Urusan Koperasi dan UKM.
- Irwanto, Djon. 2006. *Perancangan Object Oriented Software dengan UML.* Yogyakarta. Andi.
- Jogiyanto. 2005. *Analisis & Desain (Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik aplikasi Bisnis).* Yogyakarta:andi.
- Kasmir. 2005. *Bank dan lembaga keuangan lainnya:* Jakarta. PT Garfindo Persada. Edisi ke-6.
- Kenneth E. Kendall. Jullie E. Kendall. 2003. “*Systems Analysis and Design*”. fifth Edition. Dialih bahasakan oleh Thamir Abdul Hafed Al-Hamdany, B.Sc, M.Sc dalam buku *Analisa dan Perancangan Sistem Jilid 1*.Penerbit : PT Prenhallindo Jakarta.
- Khoiruddin, Arwan Ahmad. 2008. *Sistem Pendukung Keputusan Kelayakan Calon Rintisan Sekolah Bertaraf Internasional dengan Metode Fuzzy Associative Memory.* Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008): 1907-5022.
- Kusumadewi, Sri. 2004. *Artificial Intelligence (Teknik dan Aplikasinya),* Yogyakarta: Graha Ilmu.
- Kusumadewi, Sri. 2009. *Penentuan Tingkat Resiko Penyakit Menggunakan Tsukamoto Fuzzy Inference System.* Seminar Nasional II: The ApplicationOf Technology Toward A Better Life.

Maryatun, Emi, Syamsul Bahri, Rida Samdara. 2009. *Aplikasi Logika Fuzzy Metode Tsukamoto Untuk Memprediksi Potensi Serangan Stroke*. Fakultas MIPA: Universitas Bengkulu.

Marquez, Javier. 2008. *An Introduction to Credit Scoring For Small and Medium Size Enterprises*.

Masruroh, Siti Ummi,. 2008. *Sistem Pendukung Keputusan (Decision Support System) Penjurusan Program Studi Pada Sekolah Menengah Atas (Studi Kasus SMAN 1 Ciputat)*, UIN Syarif Hidayatullah Jakarta.

McLeod, Raymond, George Schell. 2004. *Sistem Informasi Manajemen Edisi Bahasa Indonesia*. Jakarta : PT. Indeks.

Muhammad. 2005. *Manajemen Pembiayaan Bank Syariah*. Yogyakarta: UPP AMP YKPN.

Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.

Munizu, Musran. 2010. *Pengaruh Faktor-Faktor Eksternal dan Internal Terhadap Kinerja Usaha Mikro dan Kecil (UMK) di Sulawesi Selatan*. Jurnal Manajemen dan Kewirausahaan, Vol.12, No.1.

Nugroho, Adi. 2005. *Analisa Perancangan Sistem Dengan Metode Berorientasi Objek*. Bandung: Informatika.

Nurdiyanto, Dodik Fredo. 2008. *Usaha Peningkatan Kualitas Pelayanan Dengan Pendekatan Fuzzy Dan Metode Service Quality Pada Pusat Perbelanjaan Assalam Hypermarket*. Universitas Muhammadiyah Surakarta.

Presman. Roger S. 2002. *Rekayasa perangkat lunak*, Yogyakarta: Andi.

- Purnomo, Herry dan Zacharias, Theo. 2004. *Pengenalan Informatika Perspektif Teknik Dan Lingkungan*. Yogyakarta: Andi.
- Rahman, Andi. 2007. *Analisa Dan Perancangan Sistem Informasi Kredit Mikro Di PT.BPR MULYA ARTA*. Program Studi Teknik Informatika: Institut Teknologi Indonesia (ITI)
- Rahmayani, Eva Yusvita. 2008. *Sistem Inferensi Fuzzy Untuk Menentukan Tingkat Resiko Penyakit Geriatri Melalui Aplikasi Berbasis Web*. Universitas Islam Indonesia: Yogyakarta.
- Republik Indonesia. 1998. Peraturan Pemerintah. Undang-undang No. 10 Tahun 1998 pasal 1 butir 12. Bank Indonesia Tahun. 1998. Jakarta.
- Republik Indonesia. 2008. Undang-undang Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil dan Menengah (UMKM). Kementerian Koperasi dan UKM. Jakarta.
- Republik Indonesia. 2008. Peraturan Menteri Keuangan no.135/PMK.05/2008 tentang fasilitas peminjaman kredit usaha rakyat menteri keuangan. Kementrian Keuangan. Jakarta.
- Sari, Eka Purnama. 2011. *Sistem Informasi Pembiayaan Mudharabah untuk Pemberian Modal bagi Usaha Kecil Menengah (UKM)*. UIN syarif Hidayatullah Jakarta.
- Situmorang, Jannes. 2005. *Kaji Tindakan Peningkatan Peran Koperasi Dan UKM Sebagai Lembaga Keuangan Alternatif*. Peneliti pada Deputi Bidang Pengkajian Sumberdaya UKMK.
- Sutrisno, Joko, Sri Lestari. 2006. *Kaji Usaha Mikro Indonesia*. Jurnal Pengkajian Koperasi dan UKM nomor 2.

Somantri, Maman, ST, MT. 2004. *Pemrograman Berorientasi Objek Menggunakan Java*. Semarang: Universitas Diponegoro.

[www.geocities.com/mmsomantri](http://www.geocities.com/mmsomantri). diakses 2 Juli 2010..

Turban, Efraim, et. al. 2005. *Decision Support System and Intelligent System (Sistem Pendukung Keputusan dan Sistem Cerdas)* Jilid 2 Ed. 7. Yogyakarta: Andi.

Wahyono, Teguh.2003. *Sistem Informasi*. Yogyakarta: Graha Ilmu..

Welling, Luke, et. Al. 2003 *PHP and MSQl Development*. Ed 2. Indianapolis: Sams Publishing.

Wibowo H, Amalia R, Fadlun A, Arivanty K. 2009. *Sistem Pendukung Keputusan untuk Menentukan Penerima Beasiswa Bank BRI Menggunakan FMADM (Studi Kasus: Mahasiswa Fakultas Teknologi Industri Universitas Islam Indonesia)*. Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009): B62- B67.

Whitten, Jeffrey L., Lonnie D. Bentley, et. al. 2004. *Metode Desain dan Analisis Sistem terjemahan dari Metode Desain dan Analisis Sistem*, 6<sup>th</sup>ed. New York- Jogyakarta : ANDI-McGraw Hill.

Widhiastiwi, Yuni. 2007. *Model Fuzzy Dengan Metode Fuzzy*. Bina Widya Vol 18 No. 02.



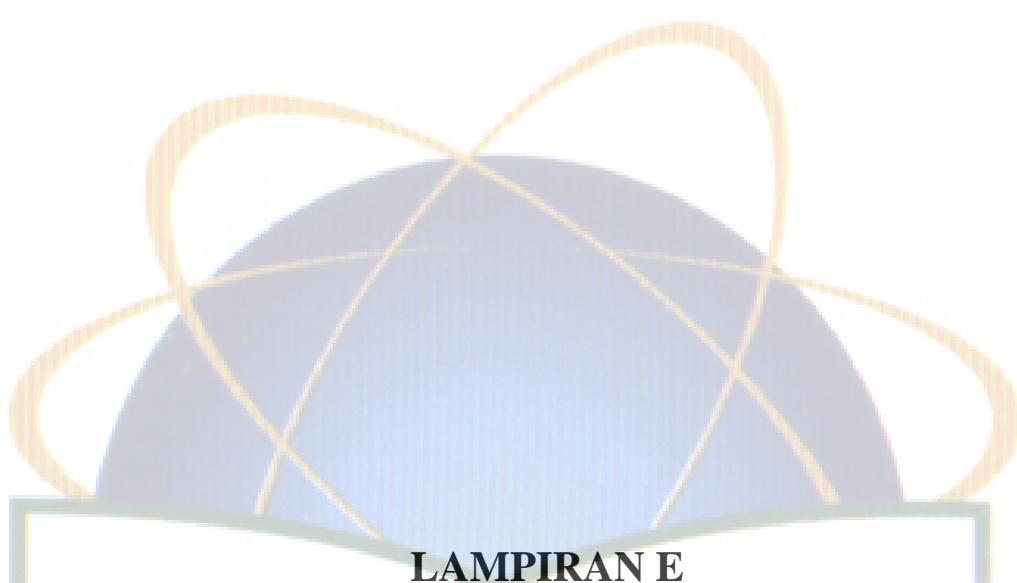






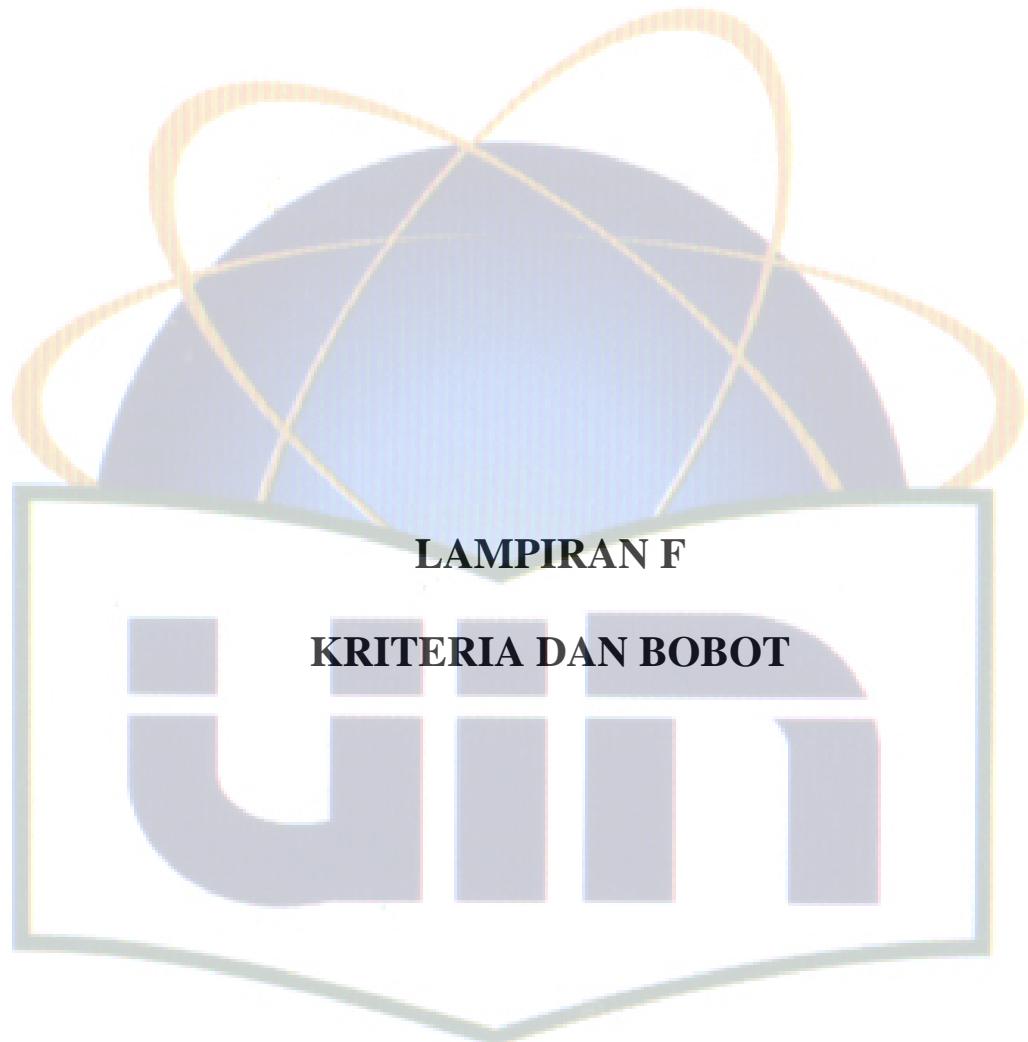
**LAMPIRAN D**

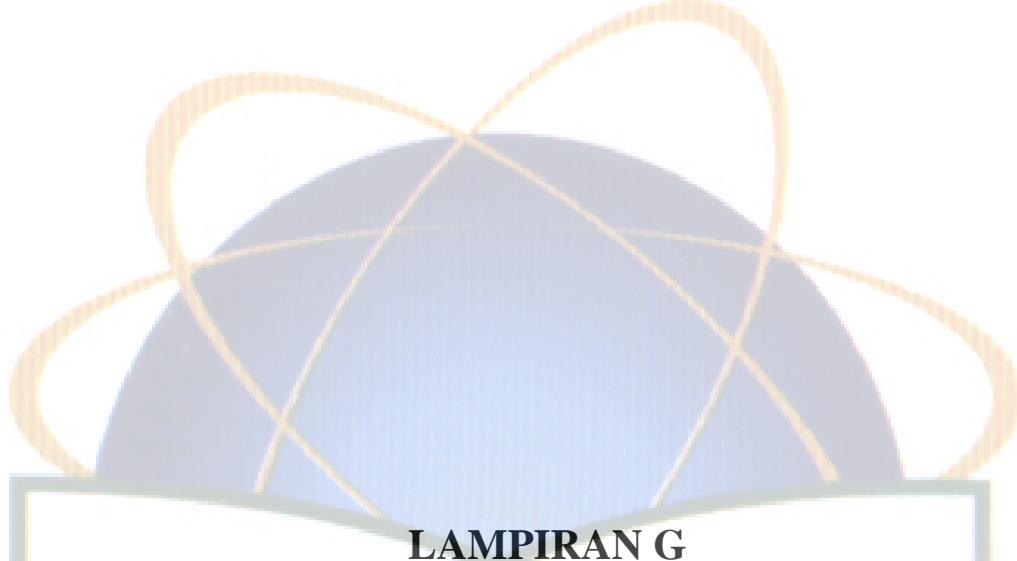
**FORMULIR PENDAFTARAN PEMBIAYAAN**



## LAMPIRAN E

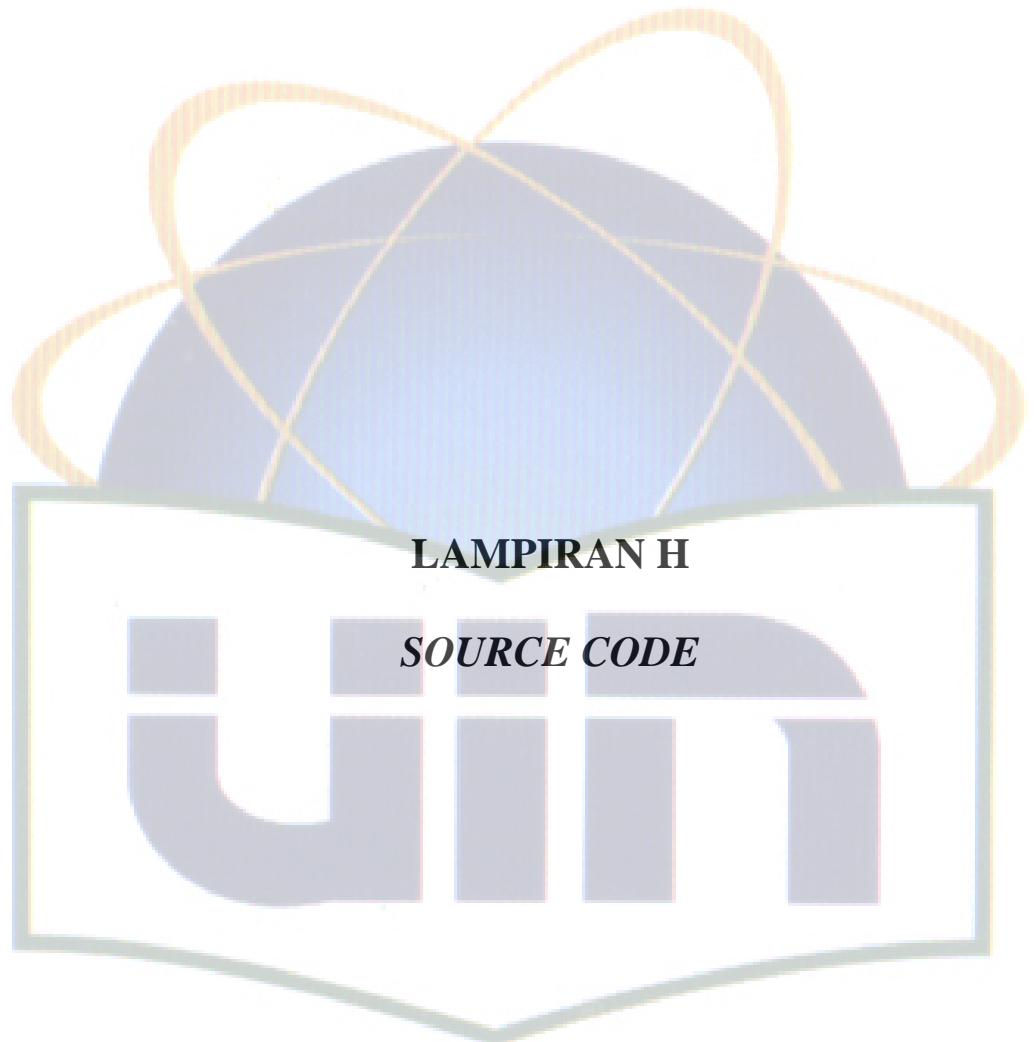
### TABEL RULE PENILAIAN KELAYAKAN PEMBIAYAAN





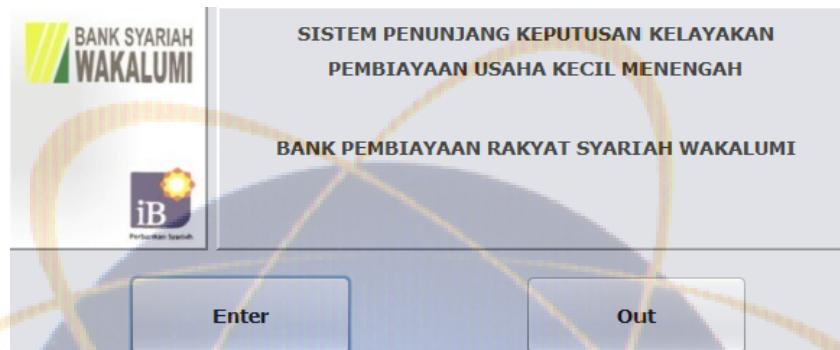
## LAMPIRAN G

### *TABEL VARIABEL INPUT DAN OUTPUT*



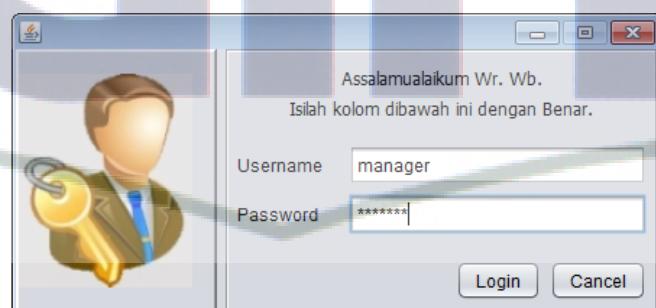
## A. Rancangan Antar Muka Admin dan Manajer

### 1. Tampilan Intro



Pada tampilan Intro terdapat dua tombol yakni enter dan out. Jika memilih tombol masuk, maka akan masuk pada menu login Sistem Penunjang Keputusan Pembiayaan BPRS dan tombol keluar untuk keluar dari program Sistem Penunjang Keputusan.

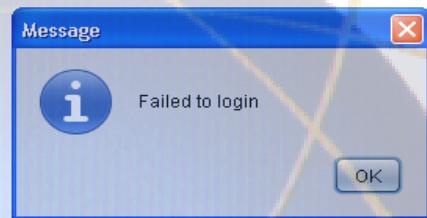
### 2. Tampilan Login



Pada Form Login ini user mengisi user name dan password sesuai dengan username dan password yang sudah di tentukan. Form Login terdiri dari dua tombol, yaitu:

1. Tombol **Login**, masukkan username dan password kemudian tekan tombol login yang sudah disediakan. Jika password yang diisikan benar maka akan muncul menu utama.

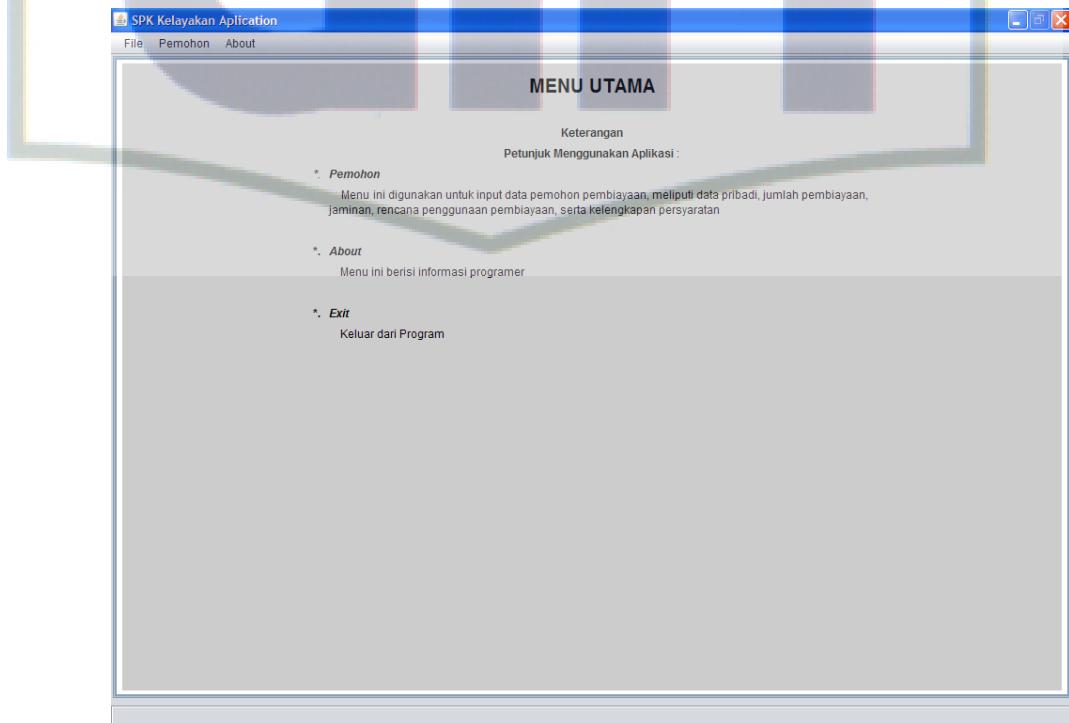
Jika username dan password yang dimasukkan salah maka akan muncul tampilan message sebagai berikut:



2. Tombol **Cancel** jika user batal untuk menjalankan sistem ini.

## B. Rancangan Antar Muka Admin

### 3. Tampilan Menu Utama



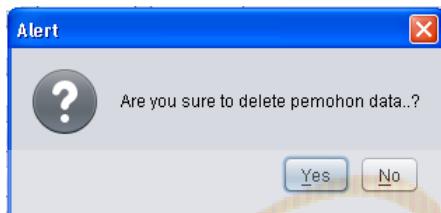
#### 4. Tampilan Menu Pemohon

Menu pemohon untuk menginput data pemohon pembiayaan ialah admin. Pada menu pemohon ini ada empat buat tombol, yakni tombol new, delete, refresh dan save.

1. **New** : tombol untuk membersihkan formulir sebelum menambah data pemohon pembiayaan.

2. **Delete** : tombol untuk menghapus data pemohon yang sudah disimpan di database.

Tampilan jendela konfirmasi jika user ingin menghapus data pemohon yang sudah disimpan



3. **Refresh** : tombol untuk membersihkan halaman formulir

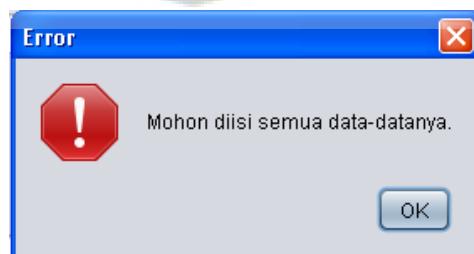
yang telah ditampilkan.

4. **Save** : tombol untuk menyimpan data setelah data di edit.

Selain empat tombol diatas, pada form menu pemohon juga menampilkan tabel dari pemohon pembiayaan yang sudah disimpan dalam database.

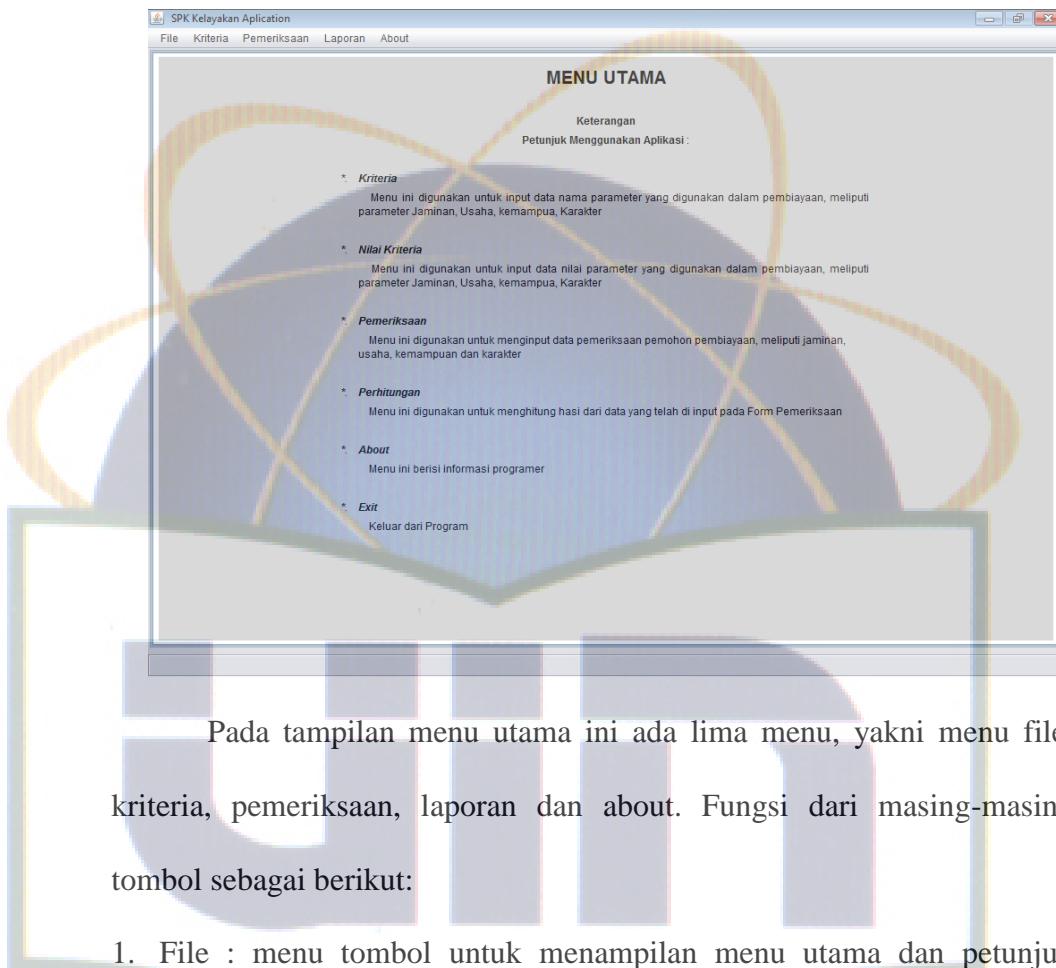
No Pemohon	Jenis Pemohon	Nama Pemohon
NSP001	Baru	Surya
NSP002	Baru	Ali
NSP003	Baru	Irwan Y
NSP004	Baru	Subri
NSP005	Baru	M. Iqbal Hadi
NSP006	Baru	Kholis

Jika data belum terisi semua maka akan tampil jendela peringatan sebagai berikut:



## C. Rancangan Antar Muka Manager

### 5. Tampilan Menu Utama

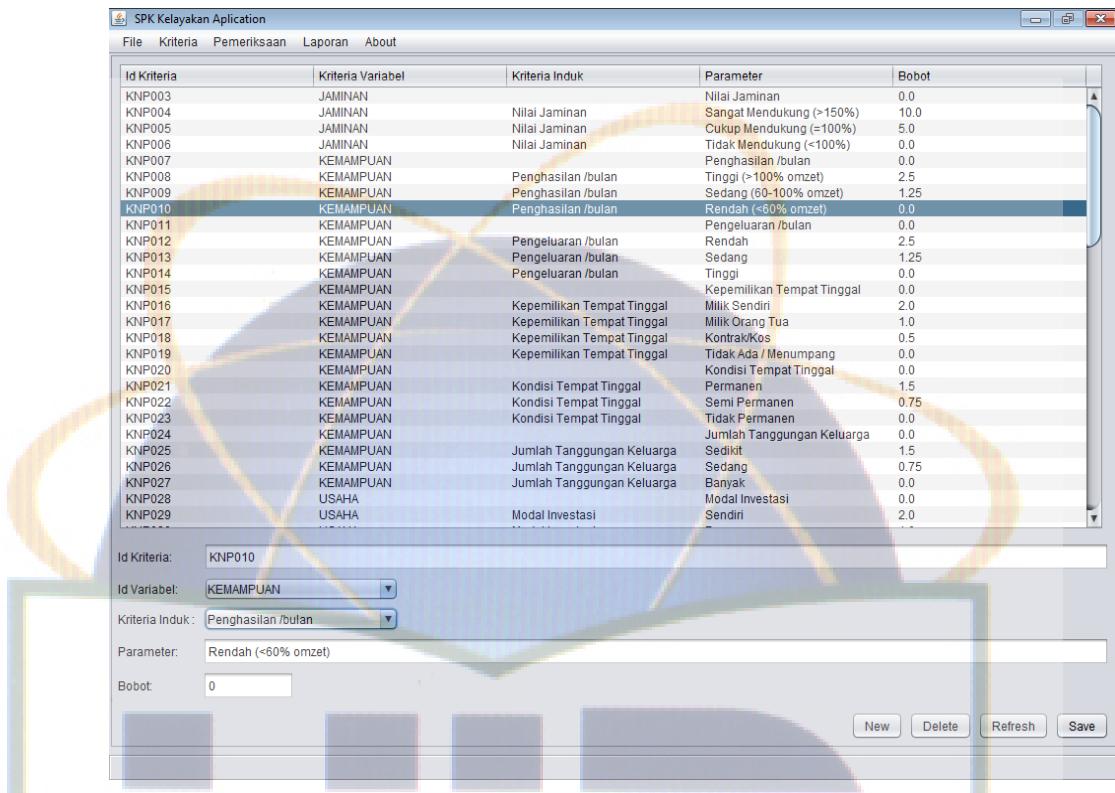


Pada tampilan menu utama ini ada lima menu, yakni menu file, kriteria, pemeriksaan, laporan dan about. Fungsi dari masing-masing tombol sebagai berikut:

1. File : menu tombol untuk menampilkan menu utama dan petunjuk penggunaan aplikasi.
2. Kriteria : menu untuk menuju ke form nama parameter dan nilai parameter.
3. Pemeriksaan : menu untuk menuju ke form Pemeriksaan
4. Laporan : menu untuk menuju ke form Laporan.
5. About : menu untuk menuju form About.

Selain lima menu diatas, pada form menu utama juga terdapat petunjuk penggunaan yang dapat dibaca langsung oleh user.

## 6. Tampilan Menu Nilai Kriteria



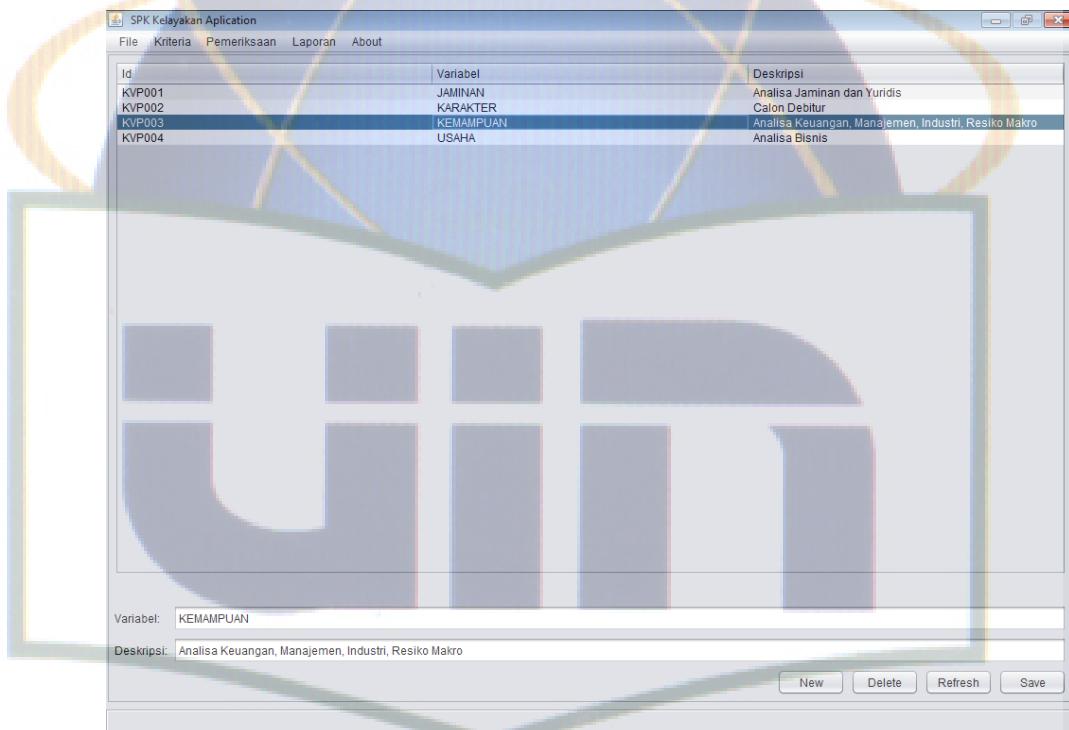
Menu Nilai Kriteria untuk mengedit rule nama parameter yang berlaku dalam perusahaan. Pada menu nilai kriteria ini ada empat buah tombol, yakni tombol New, Delete, Refresh dan Save. Fungsi dari masing-masing tombol adalah sebagai berikut:

1. **New** = Tombol untuk membersihkan formulir sebelum edit nilai parameter pembiayaan.
2. **Delete** = Tombol untuk menghapus data nilai kriteria yang terdapat parameter yang sudah disimpan di database.
3. **Refresh** = Tombol untuk membersihkan kolom formulir yang telah diedit.

4.  = tombol untuk menyimpan data setelah data dan nilai di edit.

Pada form Nilai Kriteria juga menampilkan tabel kondisi dari parameter dan nilai yang berlaku dalam setiap kriteria yang telah disimpan dalam database.

## 7. Tampilan Menu Kriteria/ Parameter



Menu Kriteria untuk mengedit nama kriteria yang terdapat pada variabel dan mendeskripsikan dari fungsi setiap variabel. Pada menu nilai kriteria ini ada empat buah tombol, yakni tombol New, Delete, Refresh dan Save. Fungsi dari masing-masing tombol adalah sebagai berikut:

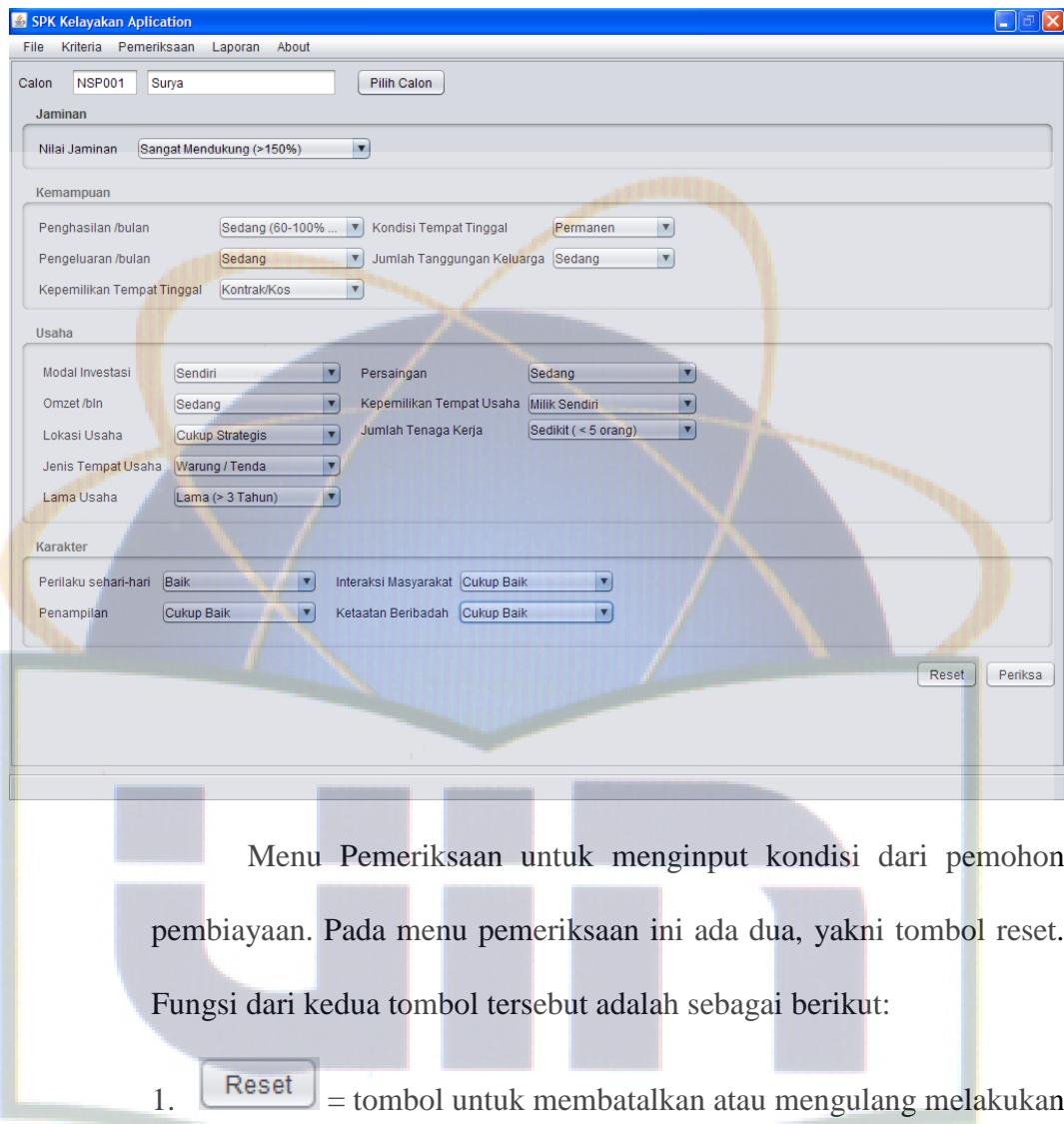
1.  = Tombol untuk membersihkan formulir sebelum menambahkan kriteria variabel ke dalam daftar nilai parameter pembiayaan.
2.  = Tombol untuk menghapus nama variabel yang terdapat parameter yang sudah disimpan di database.
3.  = Tombol untuk membersihkan kolom formulir yang telah diedit.
4.  = tombol untuk menyimpan data setelah nambah maupun edit.

Pada form Kriteria juga menampilkan tabel kondisi dari parameter, nama dan deskripsi dari kriteria yang berlaku dalam setiap kriteria yang telah disimpan dalam database.

#### 8. Tampilan Menu Pemeriksaan

Jika data kondisi kriteria belum terisi semua maka akan tampil jendela peringatan sebagai berikut:





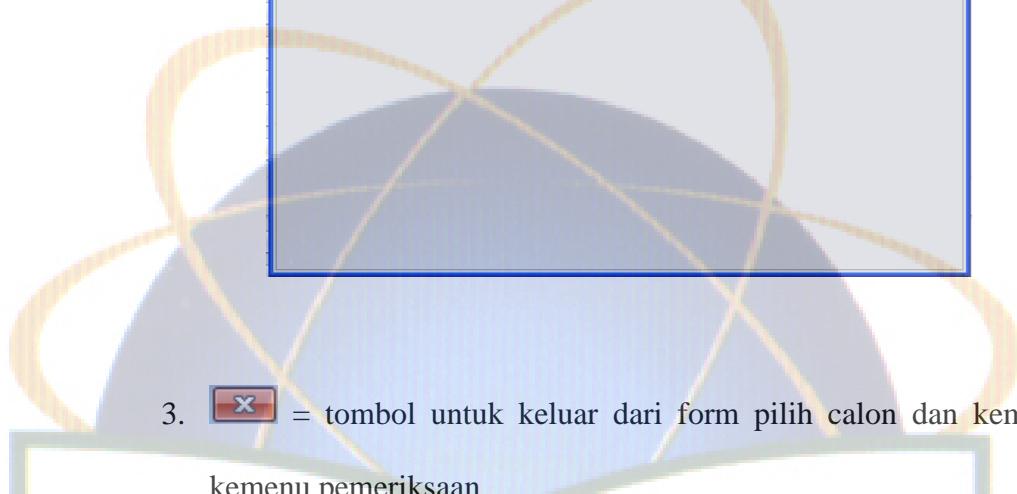
Menu Pemeriksaan untuk menginput kondisi dari pemohon pemberian. Pada menu pemeriksaan ini ada dua, yakni tombol reset.

Fungsi dari kedua tombol tersebut adalah sebagai berikut:

1. **Reset** = tombol untuk membatalkan atau mengulang melakukan pemeriksaan ulang.

2. **Periksa** = tombol untuk menambahkan data pemeriksaan dan menampilkan form perhitungan hasil penilaian kelayakan setelah kondisi dari pemohon pemberian selesai di input.

Tampilan jendela dari calon pemohon yang akan di periksa.



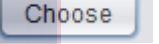
A screenshot of a Windows-style application window titled "Pilih Calon". The window contains a table with four columns: No Pemohon, Jenis Pemohon, Nama Pemohon, and Alamat. The data is as follows:

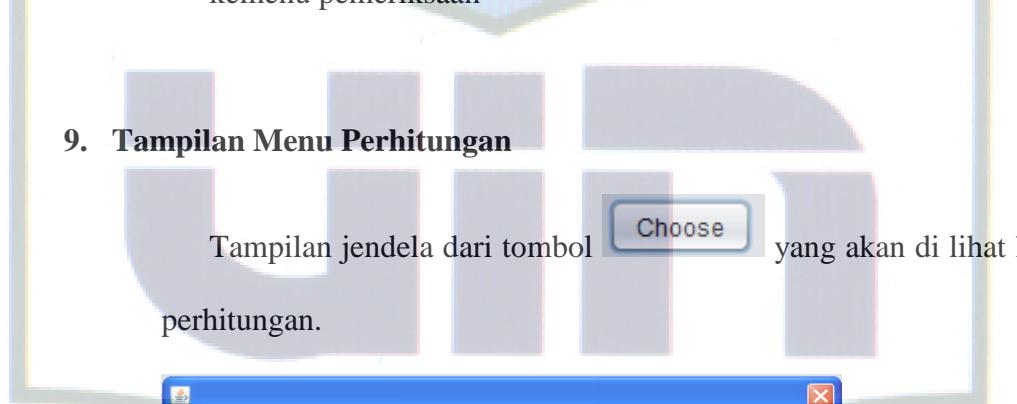
No Pemohon	Jenis Pemohon	Nama Pemohon	Alamat
NSP001	Baru	Surya	Kwitang
NSP002	Baru	Ali	Serengseng
NSP003	Baru	Irwan Y	Kampu Double Click to Choose Pemohon
NSP004	Baru	Subri	Ciputat Raya
NSP005	Baru	M. Iqbal Hadi	Jalan. sukabumi selatan no. 5 Jakarta Barat
NSP006	Baru	Kholis	Jalan peninggaran no.89 Jakarta Selatan

3.  = tombol untuk keluar dari form pilih calon dan kembali

kemenu pemeriksaan

### 9. Tampilan Menu Perhitungan

Tampilan jendela dari tombol  yang akan di lihat hasil perhitungan.



A screenshot of a Windows-style application window titled "Perhitungan". The window contains a table with four columns: ID Perhitungan, No Pemohon, Nama Pemohon, and Tanggal Perhitungan. The data is as follows:

ID Perhitungan	No Pemohon	Nama Pemohon	Tanggal Perhitungan
MPP015	NSP002	Ali	Wed May 11 15:16:42 ICT 2011
MPP017	NSP003	Irwan Y.	Wed May 11 15:23:07 ICT 2011
MPP018	NSP004	Subri	Wed May 11 15:27:29 ICT 2011
MPP019	NSP005	M. Iqbal Hadi	Wed May 11 15:34:44 ICT 2011
MPP020	NSP006	Kholis	Wed May 11 15:38:55 ICT 2011
MPP021	NSP001	Surya	Fri May 13 18:52:55 ICT 2011

**SPK Kelayakan Application**

File   Kriteria   Pemeriksaan   Laporan   About

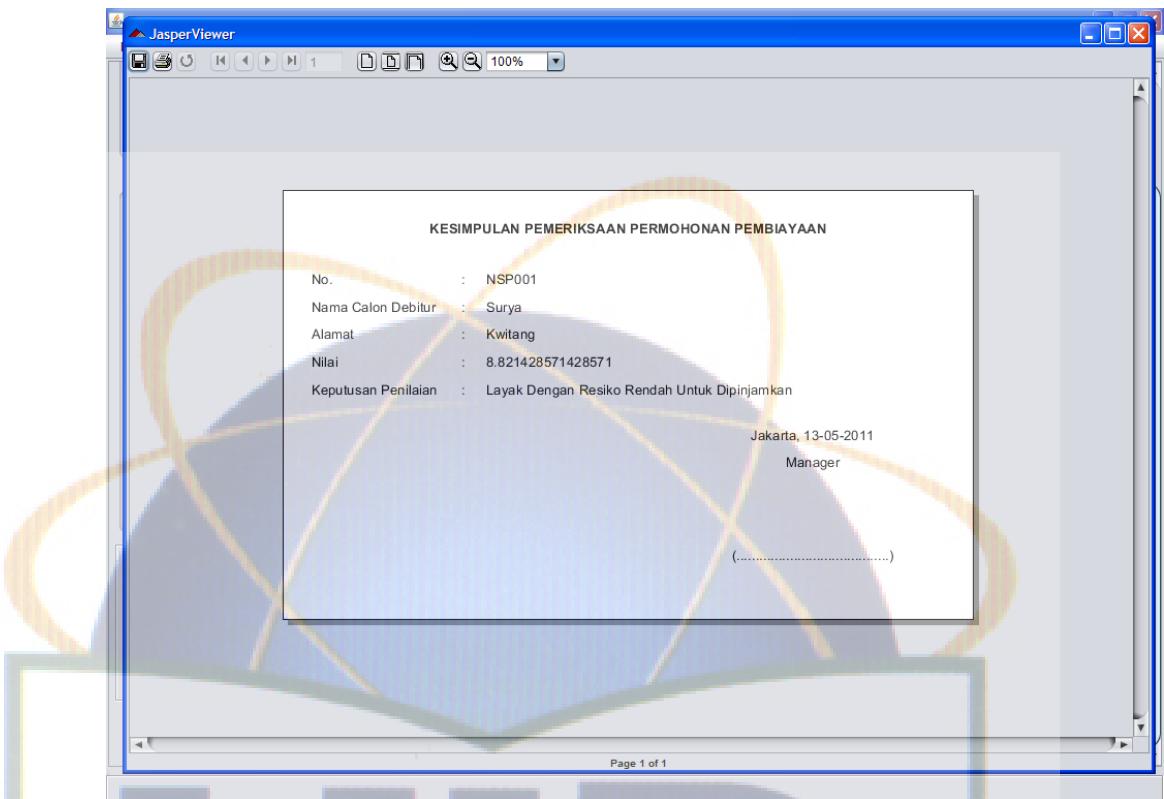
Calon NSP001   Surya   Choose

Date Jumat, Mei 13, 2011 - 6:47:03 PM

<b>Jaminan</b>	<b>Kemampuan</b>
Nilai Jaminan 10,00	Penghasilan /bulan 1,25
Jumlah Jaminan 10,00	Pengeluaran /bulan 1,25
Kepemilikan Tempat Tinggal 0,50	
Kondisi Tempat Tinggal 1,50	
Jumlah Tanggungan Keluarga 0,75	
<b>Jumlah Kemampuan 5,25</b>	
<b>Usaha</b>	<b>Karakter</b>
Modal Investasi 2,00	Perilaku sehari-hari 1,25
Omzet /bln 1,00	Penampilan 1,25
Lokasi Usaha 0,75	Interaksi Masyarakat 1,25
Jenis Tempat Usaha 0,50	Ketaatan Beribadah 2,50
Lama Usaha 1,00	<b>Jumlah Karakter 6,25</b>
Persaingan 0,50	
Kepemilikan Tempat Usaha 0,75	
Jumlah Tenaga Kerja 0,25	
<b>Jumlah Usaha 6,75</b>	
<b>Hasil Penilaian 8,821428571428571</b>	
Rekomendasi Penilaian Layak Dengan Resiko Rendah Untuk Dipinjamkan	
Jumlah Pinjaman Yang akan dipinjamkan 50.000.000	

Menu perhitungan untuk menghasilkan penilaian kelayakan dari pemohon pembiayaan. Pada menu pemeriksaan ini ada satu buah tombol, yakni tombol **Print** yang berfungsi untuk mencetak laporan hasil penilaian.

Seperti gambar dibawah ini hasil print out penilaian keputusan.



## 10. Tampilan Laporan

The screenshot shows a window titled 'SPK Kelayakan Application' with a blue header bar and a white content area. The menu bar includes File, Kriteria, Pemeriksaan, Laporan, and About. The main content is a table with the following data:

Nc Pemohon	Jenis Pemohon	Nama Pemohon	Nilai Pinjaman	Iaasi Perhitungan	Rekomendasi
NSP001	Baru	Surya	50.000.000,00	0.021420571420571	Layak Dengan Resiko Rendah Untuk Dipinjamkan
NSP002	Baru	All	20.000.000,00	5.3125	Layak Dengan Resiko Tinggi Untuk Dipinjamkan
NSP003	Baru	Irwan Y	50.000.000,00	8.791000000000000	Layak Dengan Resiko Rendah Untuk Dipinjamkan
NSP004	Baru	Subri	10.000.000,00	3.53125	Kurang Layak Untuk Dipinjamkan
NSP005	Baru	M. Iqbal Hadi	30.000.000,00	0.625	Siaga: Tidak Layak Untuk Dipinjamkan
NSP006	Baru	Kholis	20.000.000,00	0.3125	Sangat Layak Untuk Dipinjamkan

Menu Laporan untuk melihat berapa banyak peminjam yang diterima maupun di tolak dan menghapus data yang telah di hitung kedalam database.

## 11. Tampilan Menu About

Tampilan Menu About berisikan menu utama sebagai berikut:



## Source Code

### 1. KriteriaNilai.java

```
package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

/**
 * * @author haris_alkwitang
 */

@Entity
@Table(name = "kriteria_nilai", catalog = "fuzzydb", schema = "")
@NamedQueries({
    @NamedQuery(name = "KriteriaNilai.findAll", query =
"SELECT k FROM KriteriaNilai k ORDER BY k.idKriteria
ASC"),
    @NamedQuery(name = "KriteriaNilai.findByIdKriteria", query
= "SELECT k FROM KriteriaNilai k WHERE k.idKriteria =
:idKriteria"),
    @NamedQuery(name = "KriteriaNilai.findByIdVariabel", query
= "SELECT k FROM KriteriaNilai k WHERE k.idVariabel =
:idVariabel"),
    @NamedQuery(name = "KriteriaNilai.findByIdKriteriaInduk",
query = "SELECT k FROM KriteriaNilai k WHERE
k.idKriteriaInduk = :idKriteriaInduk"),
    @NamedQuery(name = "KriteriaNilai.findByParameter", query
= "SELECT k FROM KriteriaNilai k WHERE k.parameter =
:parameter"),
    @NamedQuery(name = "KriteriaNilai.findByBobot", query =
"SELECT k FROM KriteriaNilai k WHERE k.bobot = :bobot"))
public class KriteriaNilai implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id_kriteria", nullable = false, length = 11)
    private String idKriteria;
    @Basic(optional = false)
    @Column(name = "id_variabel", nullable = false, length = 11)
    private String idVariabel;
    @Column(name = "id_kriteria_induk", length = 11)
    private String idKriteriaInduk;
    @Basic(optional = false)
    @Column(name = "parameter", nullable = false, length = 100)
    private String parameter;
    @Basic(optional = false)
    @Column(name = "bobot", nullable = false)
    private double bobot;

    public KriteriaNilai() {
    }

    public KriteriaNilai(String idKriteria) {
        this.idKriteria = idKriteria;
    }

    public KriteriaNilai(String idKriteria, String idVariabel, String
parameter, double bobot) {
        this.idKriteria = idKriteria;
        this.idVariabel = idVariabel;
        this.parameter = parameter;
        this.bobot = bobot;
    }

    public String getIdKriteria() {
        return idKriteria;
    }

    public void setIdKriteria(String idKriteria) {
        this.idKriteria = idKriteria;
    }
```

```
}

    public String getIdVariabel() {
        return idVariabel;
    }

    public void setIdVariabel(String idVariabel) {
        this.idVariabel = idVariabel;
    }

    public String getIdKriteriaInduk() {
        return idKriteriaInduk;
    }

    public void setIdKriteriaInduk(String idKriteriaInduk) {
        this.idKriteriaInduk = idKriteriaInduk;
    }

    public String getParameter() {
        return parameter;
    }

    public void setParameter(String parameter) {
        this.parameter = parameter;
    }

    public double getBobot() {
        return bobot;
    }

    public void setBobot(double bobot) {
        this.bobot = bobot;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idKriteria != null ? idKriteria.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
        fields are not set
        if (!(object instanceof KriteriaNilai)) {
            return false;
        }
        KriteriaNilai other = (KriteriaNilai) object;
        if ((this.idKriteria == null && other.idKriteria != null) ||
(this.idKriteria != null &&
!this.idKriteria.equals(other.idKriteria))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "org.uin.spk.entity.KriteriaNilai[idKriteria=" +
idKriteria + "]";
    }
}
```

### 2. KriteriaVariabel.java

```
package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

/**
 * * @author haris_alkwitang
 */

@Entity
@Table(name = "kriteria_variabel", catalog = "fuzzydb", schema =
(""))
@NamedQueries({
```

```

    @NamedQuery(name = "KriteriaVariabel.findAll", query =
"SELECT k FROM KriteriaVariabel k ORDER BY k.id ASC"),
    @NamedQuery(name = "KriteriaVariabel.findById", query =
"SELECT k FROM KriteriaVariabel k WHERE k.id = :id"),
    @NamedQuery(name = "KriteriaVariabel.findByVariabel",
query = "SELECT k FROM KriteriaVariabel k WHERE k.variabel
= :variabel"),
    @NamedQuery(name = "KriteriaVariabel.findByDeskripsi",
query = "SELECT k FROM KriteriaVariabel k WHERE
k.deskripsi = :deskripsi"))
public class KriteriaVariabel implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id", nullable = false, length = 11)
    private String id;
    @Basic(optional = false)
    @Column(name = "variabel", nullable = false, length = 50)
    private String variabel;
    @Column(name = "deskripsi", length = 250)
    private String deskripsi;

    public KriteriaVariabel() {
    }
    public KriteriaVariabel(String id) {
        this.id = id;
    }
    public KriteriaVariabel(String id, String variabel) {
        this.id = id;
        this.variabel = variabel;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getVariabel() {
        return variabel;
    }
    public void setVariabel(String variabel) {
        this.variabel = variabel;
    }
    public String getDeskripsi() {
        return deskripsi;
    }
    public void setDeskripsi(String deskripsi) {
        this.deskripsi = deskripsi;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof KriteriaVariabel)) {
            return false;
        }
        KriteriaVariabel other = (KriteriaVariabel) object;
        if ((this.id == null && other.id != null) || (this.id != null &&
!this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "org.uin.spk.entity.KriteriaVariabel[id=" + id + "]";
    }
}

```

### 3. Login.java

```

package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;

/**
 * * * @author haris_alkwitang
 */

@Entity
@Table(name = "login", catalog = "fuzzydb", schema = "",
uniqueConstraints = {
    @UniqueConstraint(columnNames = {"username"})})
@NamedQueries({
    @NamedQuery(name = "Login.findAll", query = "SELECT l
FROM Login l"),
    @NamedQuery(name = "Login.findById", query = "SELECT l
FROM Login l WHERE l.id = :id"),
    @NamedQuery(name = "Login.findByUsername", query =
"SELECT l FROM Login l WHERE l.username = :username"),
    @NamedQuery(name = "Login.findByPassword", query =
"SELECT l FROM Login l WHERE l.password = :password"),
    @NamedQuery(name = "Login.findByRule", query = "SELECT l
FROM Login l WHERE l.rule = :rule"))
public class Login implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id", nullable = false, length = 11)
    private String id;
    @Basic(optional = false)
    @Column(name = "username", nullable = false, length = 50)
    private String username;
    @Basic(optional = false)
    @Column(name = "password", nullable = false, length = 50)
    private String password;
    @Basic(optional = false)
    @Column(name = "rule", nullable = false, length = 20)
    private String rule;

    public Login() {
    }
    public Login(String id) {
        this.id = id;
    }
    public Login(String id, String username, String password, String
rule) {
        this.id = id;
        this.username = username;
        this.password = password;
        this.rule = rule;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getRule() {
        return rule;
    }
}

```

```

    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getRule() {
        return rule;
    }
    public void setRule(String rule) {
        this.rule = rule;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
        fields are not set
        if (!(object instanceof Login)) {
            return false;
        }
        Login other = (Login) object;
        if ((this.id == null && other.id != null) || (this.id != null &&
        !this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "org.uin.spk.entity.Login[id=" + id + "]";
    }
}

4. MasterPerhitungan.java
package org.uin.spk.entity;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * * @author haris_alkwitang
 */
@Entity
@Table(name = "master_perhitungan", catalog = "fuzzydb",
schema = "")
@NamedQueries({
    @NamedQuery(name = "MasterPerhitungan.findAll", query =
"SELECT m FROM MasterPerhitungan m ORDER BY
m.noCalonDebitur ASC"),
    @NamedQuery(name = "MasterPerhitungan.findAll2", query =
"SELECT m FROM MasterPerhitungan m ORDER BY
m.idMasterPerhitungan ASC"),
    @NamedQuery(name =
"MasterPerhitungan.findByIdMasterPerhitungan", query =
"SELECT m FROM MasterPerhitungan m WHERE
m.idMasterPerhitungan = :idMasterPerhitungan"),
    @NamedQuery(name =
"MasterPerhitungan.findByNoCalonDebitur", query = "SELECT
m FROM MasterPerhitungan m WHERE m.noCalonDebitur =
:noCalonDebitur"),
    @NamedQuery(name =
"MasterPerhitungan.findByName", query = "SELECT
m FROM MasterPerhitungan m WHERE m.name = :name"),
    @NamedQuery(name =
"MasterPerhitungan.findByDateCheck", query = "SELECT
m FROM MasterPerhitungan m WHERE
m.dateCheck = :dateCheck"),
    @NamedQuery(name =
"MasterPerhitungan.findByNoCalonDebiturAndDateCheck",
query = "SELECT m FROM MasterPerhitungan m WHERE
m.dateCheck = :dateCheck AND m.noCalonDebitur =
:noCalonDebitur"))
public class MasterPerhitungan implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id_master_perhitungan", nullable = false,
length = 11)
    private String idMasterPerhitungan;
    @Basic(optional = false)
    @Column(name = "no_calon_debitur", nullable = false, length =
20)
    private String noCalonDebitur;
    @Basic(optional = false)
    @Column(name = "date_check", nullable = false)
    @Temporal(TemporalType.TIMESTAMP)
    private Date dateCheck;

    public MasterPerhitungan() {
    }
    public MasterPerhitungan(String idMasterPerhitungan) {
        this.idMasterPerhitungan = idMasterPerhitungan;
    }
    public MasterPerhitungan(String idMasterPerhitungan, String
noCalonDebitur, Date dateCheck) {
        this.idMasterPerhitungan = idMasterPerhitungan;
        this.noCalonDebitur = noCalonDebitur;
        this.dateCheck = dateCheck;
    }
    public String getIdMasterPerhitungan() {
        return idMasterPerhitungan;
    }
    public void setIdMasterPerhitungan(String
idMasterPerhitungan) {
        this.idMasterPerhitungan = idMasterPerhitungan;
    }
    public String getNoCalonDebitur() {
        return noCalonDebitur;
    }
    public void setNoCalonDebitur(String noCalonDebitur) {
        this.noCalonDebitur = noCalonDebitur;
    }
    public Date getDateCheck() {
        return dateCheck;
    }
    public void setDateCheck(Date dateCheck) {
        this.dateCheck = dateCheck;
    }
    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idMasterPerhitungan != null ?
idMasterPerhitungan.hashCode() : 0);
        return hash;
    }
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
        fields are not set
        if (!(object instanceof MasterPerhitungan)) {
            return false;
        }
        MasterPerhitungan other = (MasterPerhitungan) object;
        if ((this.idMasterPerhitungan == null &&
other.idMasterPerhitungan != null) || (this.idMasterPerhitungan !=
null &&
!this.idMasterPerhitungan.equals(other.idMasterPerhitungan))) {
            return false;
        }
        return true;
    }
}

```

```

    }

    @Override
    public String toString() {
        return
"org.uin.spk.entity.MasterPerhitungan[idMasterPerhitungan=" +
idMasterPerhitungan + "];"
    }
}

```

## 5. Pemohon.java

```

package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

/**
 *
 * @author haris_alkwitang
 */
@Entity
@Table(name = "pemohon", catalog = "fuzzydb", schema = "")
@NamedQueries({
    @NamedQuery(name = "Pemohon.findAll", query = "SELECT
p FROM Pemohon p ORDER BY p.noCalonDebitur ASC"),
    @NamedQuery(name = "Pemohon.findByNoCalonDebitur",
query = "SELECT p FROM Pemohon p WHERE
p.noCalonDebitur = :noCalonDebitur"),
    @NamedQuery(name = "Pemohon.findByJenisPemohon",
query = "SELECT p FROM Pemohon p WHERE p.jenisPemohon =
:jenisPemohon"),
    @NamedQuery(name = "Pemohon.findByNamaCalonDebitur",
query = "SELECT p FROM Pemohon p WHERE
p.namaCalonDebitur = :namaCalonDebitur"),
    @NamedQuery(name = "Pemohon.findByAlamat",
query = "SELECT p FROM Pemohon p WHERE p.alamat = :alamat"),
    @NamedQuery(name = "Pemohon.findByNoTelp",
query = "SELECT p FROM Pemohon p WHERE p.noTelp = :noTelp"),
    @NamedQuery(name = "Pemohon.findByPekerjaan",
query = "SELECT p FROM Pemohon p WHERE p.pekerjaan =
:pekerjaan"),
    @NamedQuery(name =
"Pemohon.findByRencanaPenggunaan", query = "SELECT p
FROM Pemohon p WHERE p.rencanaPenggunaan =
:rencanaPenggunaan"),
    @NamedQuery(name = "Pemohon.findByNilaiJaminan",
query = "SELECT p FROM Pemohon p WHERE p.nilaiJaminan =
:nilaiJaminan"),
    @NamedQuery(name = "Pemohon.findByJangkaWaktuBulan",
query = "SELECT p FROM Pemohon p WHERE
p.jangkaWaktuBulan = :jangkaWaktuBulan"),
    @NamedQuery(name =
"Pemohon.findByJangkaWaktuRupiah", query = "SELECT p
FROM Pemohon p WHERE p.jangkaWaktuRupiah =
:jangkaWaktuRupiah"),
    @NamedQuery(name = "Pemohon.findByJenisJaminan",
query = "SELECT p FROM Pemohon p WHERE p.jenisJaminan =
:jenisJaminan"),
    @NamedQuery(name = "Pemohon.findByJenisJaminanThn",
query = "SELECT p FROM Pemohon p WHERE
p.jenisJaminanThn = :jenisJaminanThn"),
    @NamedQuery(name = "Pemohon.findByNilaiTaksasi",
query = "SELECT p FROM Pemohon p WHERE p.nilaiTaksasi =
:nilaiTaksasi"),
    @NamedQuery(name = "Pemohon.findByAdministrasi",
query = "SELECT p FROM Pemohon p WHERE p.administrasi =
:administrasi"),
    @NamedQuery(name = "Pemohon.findByAkad",
query = "SELECT p FROM Pemohon p WHERE p.akad = :akad"))
}

```

```

public class Pemohon implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "no_calon_debitur", nullable = false, length =
20)
    private String noCalonDebitur;
    @Basic(optional = false)
    @Column(name = "jenis_pemohon", nullable = false, length =
50)
    private String jenisPemohon;
    @Basic(optional = false)
    @Column(name = "nama_calon_debitur", nullable = false,
length = 30)
    private String namaCalonDebitur;
    @Basic(optional = false)
    @Column(name = "alamat", nullable = false, length = 50)
    private String alamat;
    @Column(name = "no_telp", length = 50)
    private String noTelp;
    @Basic(optional = false)
    @Column(name = "pekerjaan", nullable = false, length = 50)
    private String pekerjaan;
    @Basic(optional = false)
    @Column(name = "rencana_penggunaan", nullable = false,
length = 50)
    private String rencanaPenggunaan;
    @Basic(optional = false)
    @Column(name = "nilai_jaminan", nullable = false, length =
15)
    private String nilaiJaminan;
    @Basic(optional = false)
    @Column(name = "jangka_waktu_bulan", nullable = false,
length = 20)
    private String jangkaWaktuBulan;
    @Column(name = "jangka_waktu_rupiah", length = 15)
    private String jangkaWaktuRupiah;
    @Basic(optional = false)
    @Column(name = "jenis_jaminan", nullable = false, length =
20)
    private String jenisJaminan;
    @Column(name = "jenis_jaminan_thn", length = 10)
    private String jenisJaminanThn;
    @Basic(optional = false)
    @Column(name = "nilai_taksasi", nullable = false, length = 15)
    private String nilaiTaksasi;
    @Basic(optional = false)
    @Column(name = "administrasi", nullable = false, length = 20)
    private String administrasi;
    @Lob
    @Column(name = "pemohonimage")
    private byte[] pemohonimage;
    @Basic(optional = false)
    @Column(name = "akad", nullable = false, length = 60)
    private String akad;

    public Pemohon() {
    }

    public Pemohon(String noCalonDebitur) {
        this.noCalonDebitur = noCalonDebitur;
    }

    public Pemohon(String noCalonDebitur, String jenisPemohon,
String namaCalonDebitur, String alamat, String pekerjaan, String
rencanaPenggunaan, String nilaiJaminan, String
jangkaWaktuBulan, String jenisJaminan, String nilaiTaksasi,
String administrasi, String akad) {
        this.noCalonDebitur = noCalonDebitur;
        this.jenisPemohon = jenisPemohon;
        this.namaCalonDebitur = namaCalonDebitur;
        this.alamat = alamat;
        this.pekerjaan = pekerjaan;
        this.rencanaPenggunaan = rencanaPenggunaan;
        this.nilaiJaminan = nilaiJaminan;
        this.jangkaWaktuBulan = jangkaWaktuBulan;
    }
}

```

```

this.jenisJaminan = jenisJaminan;
this.nilaiTaksasi = nilaiTaksasi;
this.administrasi = administrasi;
this.akad = akad;
}

public String getNoCalonDebitur() {
    return noCalonDebitur;
}
public void setNoCalonDebitur(String noCalonDebitur) {
    this.noCalonDebitur = noCalonDebitur;
}
public String getJenisPemohon() {
    return jenisPemohon;
}
public void setJenisPemohon(String jenisPemohon) {
    this.jenisPemohon = jenisPemohon;
}
public String getNamaCalonDebitur() {
    return namaCalonDebitur;
}
public void setNamaCalonDebitur(String namaCalonDebitur) {
    this.namaCalonDebitur = namaCalonDebitur;
}
public String getAlamat() {
    return alamat;
}
public void setAlamat(String alamat) {
    this.alamat = alamat;
}
public String getNoTelp() {
    return noTelp;
}
public void setNoTelp(String noTelp) {
    this.noTelp = noTelp;
}
public String getPekerjaan() {
    return pekerjaan;
}
public void setPekerjaan(String pekerjaan) {
    this.pekerjaan = pekerjaan;
}
public String getRencanaPenggunaan() {
    return rencanaPenggunaan;
}
public void setRencanaPenggunaan(String rencanaPenggunaan) {
    this.rencanaPenggunaan = rencanaPenggunaan;
}
public String getNilaiJaminan() {
    return nilaiJaminan;
}
public void setNilaiJaminan(String nilaiJaminan) {
    this.nilaiJaminan = nilaiJaminan;
}
public String getJangkaWaktuBulan() {
    return jangkaWaktuBulan;
}
public void setJangkaWaktuBulan(String jangkaWaktuBulan) {
    this.jangkaWaktuBulan = jangkaWaktuBulan;
}
public String getJangkaWaktuRupiah() {
    return jangkaWaktuRupiah;
}
public void setJangkaWaktuRupiah(String jangkaWaktuRupiah) {
    this.jangkaWaktuRupiah = jangkaWaktuRupiah;
}
public String getJenisJaminan() {
    return jenisJaminan;
}
public void setJenisJaminan(String jenisJaminan) {
    this.jenisJaminan = jenisJaminan;
}

public String getJenisJaminanThn() {
    return jenisJaminanThn;
}
public void setJenisJaminanThn(String jenisJaminanThn) {
    this.jenisJaminanThn = jenisJaminanThn;
}
public String getNilaiTaksasi() {
    return nilaiTaksasi;
}
public void setNilaiTaksasi(String nilaiTaksasi) {
    this.nilaiTaksasi = nilaiTaksasi;
}
public String getAdministrasi() {
    return administrasi;
}
public void setAdministrasi(String administrasi) {
    this.administrasi = administrasi;
}
public byte[] getPemohonimage() {
    return pemohonimage;
}
public void setPemohonimage(byte[] pemohonimage) {
    this.pemohonimage = pemohonimage;
}
public String getAkad() {
    return akad;
}
public void setAkad(String akad) {
    this.akad = akad;
}
@Override
public int hashCode() {
    int hash = 0;
    hash += (noCalonDebitur != null ?
noCalonDebitur.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id
    fields are not set
    if (!(object instanceof Pemohon)) {
        return false;
    }
    Pemohon other = (Pemohon) object;
    if ((this.noCalonDebitur == null && other.noCalonDebitur !=
null) || (this.noCalonDebitur != null &&
!this.noCalonDebitur.equals(other.noCalonDebitur))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "org.uin.spk.entity.Pemohon[noCalonDebitur=" +
noCalonDebitur + "]";
}
}


```

**6. Perhitungan.java**

```

package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
/**
 * * * @author haris_alkwitang

```

```

*/
@Entity
@Table(name = "perhitungan", catalog = "fuzzydb", schema = "")
@NamedQueries({
    @NamedQuery(name = "Perhitungan.findAll", query =
"SELECT p FROM Perhitungan p"),
    @NamedQuery(name =
"Perhitungan.findByIdMasterPerhitungan", query = "SELECT p
FROM Perhitungan p WHERE
p.perhitunganPK.idMasterPerhitungan = :idMasterPerhitungan"),
    @NamedQuery(name = "Perhitungan.findByIdKriteria", query =
"SELECT p FROM Perhitungan p WHERE
p.perhitunganPK.idKriteria = :idKriteria"),
    @NamedQuery(name = "Perhitungan.findByNilai", query =
"SELECT p FROM Perhitungan p WHERE p.nilai = :nilai"))
public class Perhitungan implements Serializable {
    private static final long serialVersionUID = 1L;
    @EmbeddedId
    protected PerhitunganPK perhitunganPK;
    @Basic(optional = false)
    @Column(name = "nilai", nullable = false)
    private double nilai;

    public Perhitungan() {
    }
    public Perhitungan(PerhitunganPK perhitunganPK) {
        this.perhitunganPK = perhitunganPK;
    }
    public Perhitungan(PerhitunganPK perhitunganPK, double nilai)
    {
        this.perhitunganPK = perhitunganPK;
        this.nilai = nilai;
    }

    public Perhitungan(String idMasterPerhitungan, String
idKriteria) {
        this.perhitunganPK = new
PerhitunganPK(idMasterPerhitungan, idKriteria);
    }
    public PerhitunganPK getPerhitunganPK() {
        return perhitunganPK;
    }
    public void setPerhitunganPK(PerhitunganPK perhitunganPK) {
        this.perhitunganPK = perhitunganPK;
    }
    public double getNilai() {
        return nilai;
    }
    public void setNilai(double nilai) {
        this.nilai = nilai;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (perhitunganPK != null ? perhitunganPK.hashCode()
: 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof Perhitungan)) {
            return false;
        }
        Perhitungan other = (Perhitungan) object;
        if ((this.perhitunganPK == null && other.perhitunganPK !=
null) || (this.perhitunganPK != null &&
!this.perhitunganPK.equals(other.perhitunganPK))) {
            return false;
        }
        return true;
    }
}

```

```

@Override
public String toString() {
    return "org.uin.spk.entity.Perhitungan[perhitunganPK=" +
perhitunganPK + "]";
}
}

```

## 7. PerhitunganPK.java

```

package org.uin.spk.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Embeddable;

/**
 * * * @author haris_alkwitang
 */

@Embeddable
public class PerhitunganPK implements Serializable {
    @Basic(optional = false)
    @Column(name = "id_master_perhitungan", nullable = false,
length = 11)
    private String idMasterPerhitungan;
    @Basic(optional = false)
    @Column(name = "id_kriteria", nullable = false, length = 11)
    private String idKriteria;

    public PerhitunganPK() {
    }
    public PerhitunganPK(String idMasterPerhitungan, String
idKriteria) {
        this.idMasterPerhitungan = idMasterPerhitungan;
        this.idKriteria = idKriteria;
    }
    public String getIdMasterPerhitungan() {
        return idMasterPerhitungan;
    }
    public void setIdMasterPerhitungan(String
idMasterPerhitungan) {
        this.idMasterPerhitungan = idMasterPerhitungan;
    }
    public String getIdKriteria() {
        return idKriteria;
    }
    public void setIdKriteria(String idKriteria) {
        this.idKriteria = idKriteria;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idMasterPerhitungan.hashCode() : 0);
        hash += (idKriteria.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof PerhitunganPK)) {
            return false;
        }
        PerhitunganPK other = (PerhitunganPK) object;
        if ((this.idMasterPerhitungan == null) || (this.idMasterPerhitungan !=
null &&
other.idMasterPerhitungan != null) || (this.idMasterPerhitungan !=
null &&
!this.idMasterPerhitungan.equals(other.idMasterPerhitungan))) {
            return false;
        }
    }
}

```

```

        if ((this.idKriteria == null && other.idKriteria != null) ||
(this.idKriteria != null &&
!this.idKriteria.equals(other.idKriteria))) {
            return false;
        }
        return true;
    }

@Override
public String toString() {
    return
"org.uin.spk.entity.PerhitunganPK[idMasterPerhitungan=" +
idMasterPerhitungan + ", idKriteria=" + idKriteria + "]";
}
}

```

## 8. FuzzyLogic.java

```

package org.uin.spk.logic;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import java.util.StringTokenizer;
import javax.swing.JOptionPane;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;


$$\text{Fuzzy Logic Diagram: A 3D surface plot showing three overlapping Gaussian membership functions (blue, red, green) defined over a 2D plane. The axes represent input variables (x, y) and output values (z). The surfaces peak at different coordinates, representing the degree of membership for each rule's premise. The blue surface is centered around (6, 6), the red around (10, 6), and the green around (10, 10).}$$



$$\text{Input Range (TIDAK\_BAGUS, CUKUP\_BAGUS, SANGAT\_BAGUS, ERROR)}$$


$$\text{Output Range (SANGAT\_TIDAK\_LAYAK, KURANG\_LAYAK, LAYAK\_DENGAN\_RESIKO\_TINGGI, LAYAK\_DENGAN\_RESIKO\_RENDAH, SANGAT\_LAYAK, ERROR)}$$


public class FuzzyLogic {

    public static enum InputRange {
        TIDAK_BAGUS, CUKUP_BAGUS, SANGAT_BAGUS,
        ERROR
    }

    public static enum OutputRange {
        SANGAT_TIDAK_LAYAK, KURANG_LAYAK,
        LAYAK_DENGAN_RESIKO_TINGGI,
        LAYAK_DENGAN_RESIKO_RENDAH, SANGAT_LAYAK,
        ERROR
    }

    public static HashMap<String, OutputRange> listOfRule = new
HashMap<String, OutputRange>();
    private String SL = Function.getResources("system.file.sl");
    private String LRR = Function.getResources("system.file.lrr");
    private String LRT = Function.getResources("system.file.lrt");
    private String KL = Function.getResources("system.file.kl");
    private String STL = Function.getResources("system.file.stl");
    private String SB = Function.getResources("system.file.sb");
    private String CB = Function.getResources("system.file.cb");
    private String TB = Function.getResources("system.file.tb");
    private String sp =
Function.getResources("system.file.secondseparator");

    public List<InputRange> checkRangeVariableInput(double x) {
        List<InputRange> listInputRange = new
ArrayList<InputRange>();
        if (x <= 10.0 && x >= 6.0) {
            listInputRange.add(InputRange.SANGAT_BAGUS);
        }
        if (x <= 7.0 && x >= 3.0) {
            listInputRange.add(InputRange.CUKUP_BAGUS);
        }
        if (x <= 4.0 && x >= 0.0) {

```

```

            listInputRange.add(InputRange.TIDAK_BAGUS);
        }
        return listInputRange;
    }

    public OutputRange checkRangeVariableOutput(double z) {
        if (9.0 < z && z <= 10) {
            return OutputRange.SANGAT_LAYAK;
        } else if (6.0 < z && z <= 9.0) {
            return
OutputRange.LAYAK_DENGAN_RESIKO_RENDAH;
        } else if (4.0 < z && z <= 6.0) {
            return
OutputRange.LAYAK_DENGAN_RESIKO_TINGGI;
        } else if (1.0 < z && z <= 4.0) {
            return OutputRange.KURANG_LAYAK;
        } else if (0.0 < z && z <= 1.0) {
            return OutputRange.SANGAT_TIDAK_LAYAK;
        } else {
            return OutputRange.ERROR;
        }
    }

    public double calculateInputIfSB(double x) {
        if (x <= 6) {
            return 0;
        } else if (x >= 6.0 && x <= 10.0) {
            return (x - 6) / (10 - 6);
        } else if (x >= 10.0) {
            return 1;
        }
        return 0.0;
    }

    public double calculateInputIfCB(double x) {
        if (x <= 3.0 || x >= 7.0) {
            return 0;
        } else if (x >= 3.0 && x <= 5.0) {
            return (x - 3) / (5 - 3);
        } else if (x >= 5.0 && x <= 7.0) {
            return (7 - x) / (7 - 5);
        }
        return 0.0;
    }

    public double calculateInputIfTB(double x) {
        if (x <= 0) {
            return 1;
        } else if (x >= 0.0 && x <= 4.0) {
            return (4 - x) / (4 - 0);
        } else if (x >= 4.0) {
            return 0;
        }
        return 0.0;
    }

    public double calculateOutputIfSL(double z) {
        if (z <= 9.0) {
            return 0;
        } else if (9.0 <= z && z <= 10.0) {
            return (z - 9) / (10 - 9);
        } else if (z >= 10.0) {
            return 1;
        }
        return 0.0;
    }

    public double getZValueIfSL(double outVal) {
        return (outVal + 9) * (10 - 9);
    }

    public double calculateOutputIfLRR(double z) {
        if (z <= 6.0 || z >= 9.0) {
            return 0;
        } else if (6.0 <= z && z <= 7.5) {
            return (z - 6) / (7.5 - 6);
        }
    }
}

```

```

    } else if (7.5 <= z && z <= 9.0) {
        return (9 - z) / (9 - 7.5);
    }
    return 0.0;
}

// public double getZValueIfLRR(double outVal) {
//     return (outVal + 6) * (7.5 - 6);
// }

public double getZValueIfLRR(double outVal) {
    return 9 - (outVal * (9 - 7.5));
}

public double calculateOutputIfLRT(double z) {
    if (z <= 4.0 || z >= 6.0) {
        return 0;
    } else if (4.0 <= z && z <= 5.0) {
        return (z - 4) / (5 - 4);
    } else if (5.0 <= z && z <= 6.0) {
        return (6 - z) / (6 - 5);
    }
    return 0.0;
}

// public double getZValueIfLRT(double outVal) {
//     return (outVal + 4) * (5 - 4);
// }

public double getZValueIfLRT(double outVal) {
    return 6 - (outVal * (6 - 5));
}

public double calculateOutputIfKL(double z) {
    if (z <= 1.0 || z >= 4.0) {
        return 0;
    } else if (1.0 <= z && z <= 2.5) {
        return (z - 1) / (2.5 - 1);
    } else if (2.5 <= z && z <= 4.0) {
        return (4 - z) / (4 - 2.5);
    }
    return 0.0;
}

// public double getZValueIfKL(double outVal) {
//     return (outVal + 1) * (2.5 - 1);
// }

public double getZValueIfKL(double outVal) {
    return 4 - (outVal * (4 - 2.5));
}

public double calculateOutputIfSTL(double z) {
    if (z <= 0.0) {
        return 1;
    } else if (0.0 <= z && z <= 1.0) {
        return (1 - z) / (1 - 0);
    } else if (z >= 1.0) {
        return 0.0;
    }
    return 0.0;
}

public double getZValueIfSTL(double outVal) {
    return (1 - outVal) * (1 - 0);
}

public double calculateBigZ(List<ZN> listZn) {
    double totalValue = 0.0;
    double pembagi = 0.0;
    for (ZN zn : listZn) {
        totalValue += zn.getAxZ();
        pembagi += zn.getA();
    }
    System.out.println("sum a x z ->\t" + totalValue);
    System.out.println("sum a ->\t" + pembagi);
}

System.out.println("result ->\t" + totalValue/pembagi);
return totalValue / pembagi;
}

public OutputRange getVariableOutput(InputRange
inputJaminan, InputRange inputUsaha, InputRange
inputKemampuan, InputRange inputKarakter) {
    String key =
getAliasRange(inputJaminan).concat(sp).concat(getAliasRange(in
putUsaha)).concat(sp);

concat(getAliasRange(inputKemampuan).concat(sp).concat(getAli
asRange(inputKarakter)));
if (listOfRule.containsKey(key)) {
    return listOfRule.get(key);
} else {
    return null;
}
}

public String getAliasRange(InputRange range) {
    if (range.equals(InputRange.SANGAT_BAGUS)) {
        return SB;
    } else if (range.equals(InputRange.CUKUP_BAGUS)) {
        return CB;
    } else if (range.equals(InputRange.TIDAK_BAGUS)) {
        return TB;
    } else {
        return "";
    }
}

public void loadRule() {
    try {
        String input;
        BufferedReader br = new BufferedReader(new
FileReader(Function.getResources("system.file.rule")));
        StringTokenizer str;
        String text;
        String temp[];
        listOfRule = new HashMap<String, OutputRange>();
        OutputRange outputRange;
        int i = 1;
        while ((input = br.readLine()) != null) {
            if (i == 1) {
                i++;
                continue;
            }
            input = input.trim();
            str = new StringTokenizer(input);
            text = str.nextToken();
            temp =
text.split(Function.getResources("system.file.separator"));
            if (temp.length == 2) {
                if (temp[1].equalsIgnoreCase(SL)) {
                    outputRange = OutputRange.SANGAT_LAYAK;
                } else if (temp[1].equalsIgnoreCase(LRR)) {
                    outputRange =
OutputRange.LAYAK_DENGAN_RISIKO_RENDAH;
                } else if (temp[1].equalsIgnoreCase(LRT)) {
                    outputRange =
OutputRange.LAYAK_DENGAN_RISIKO_TINGGI;
                } else if (temp[1].equalsIgnoreCase(KL)) {
                    outputRange = OutputRange.KURANG_LAYAK;
                } else if (temp[1].equalsIgnoreCase(STL)) {
                    outputRange =
OutputRange.SANGAT_TIDAK_LAYAK;
                } else {
                    outputRange = OutputRange.ERROR;
                }
                listOfRule.put(temp[0], outputRange);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        }

    public HashMap<InputRange, Double>
    calculateRangeVariableInput(List<InputRange> listInputRange,
double inputVal) {
    HashMap<InputRange, Double> returnMap = new
    HashMap<InputRange, Double>();
    for (InputRange inputRange : listInputRange) {
        if (inputRange.equals(InputRange.SANGAT_BAGUS)) {
            returnMap.put(InputRange.SANGAT_BAGUS,
Double.valueOf(calculateInputIfSB(inputVal)));
        } else if
        (inputRange.equals(InputRange.CUKUP_BAGUS)) {
            returnMap.put(InputRange.CUKUP_BAGUS,
Double.valueOf(calculateInputIfCB(inputVal)));
        } else if
        (inputRange.equals(InputRange.TIDAK_BAGUS)) {
            returnMap.put(InputRange.TIDAK_BAGUS,
Double.valueOf(calculateInputIfTB(inputVal)));
        }
    }
    return returnMap;
}

public double calculate(Kriteria initKriteria) {
    if(listOfRule.size()==0){
        loadRule();
    }

    SingleKriteria jaminanKriteria = new SingleKriteria();
    jaminanKriteria.setKriteriaValue(initKriteria.getJaminan());
    jaminanKriteria.setRangeKriteriaInputValue(checkRangeVariableI
nput(jaminanKriteria.getKriteriaValue()));

    jaminanKriteria.setKriteriaInputValue(calculateRangeVariableInpu
t(jaminanKriteria.getRangeKriteriaInputValue(),
jaminanKriteria.getKriteriaValue()));

    SingleKriteria usahaKriteria = new SingleKriteria();
    usahaKriteria.setKriteriaValue(initKriteria.getUsaha());
    usahaKriteria.setRangeKriteriaInputValue(checkRangeVariableInpu
t(usahaKriteria.getKriteriaValue()));

    usahaKriteria.setKriteriaInputValue(calculateRangeVariableInput(
usaKriteria.getRangeKriteriaInputValue(),
usaKriteria.getKriteriaValue()));

    SingleKriteria kemampuanKriteria = new SingleKriteria();
    kemampuanKriteria.setKriteriaValue(initKriteria.getKemampuan());
    kemampuanKriteria.setRangeKriteriaInputValue(checkRangeVaria
bleInput(kemampuanKriteria.getKriteriaValue()));

    kemampuanKriteria.setKriteriaInputValue(calculateRangeVariable
Input(kemampuanKriteria.getRangeKriteriaInputValue(),
kemampuanKriteria.getKriteriaValue()));

    SingleKriteria karakterKriteria = new SingleKriteria();
    karakterKriteria.setKriteriaValue(initKriteria.getKarakter());
    karakterKriteria.setRangeKriteriaInputValue(checkRangeVariableI
nput(karakterKriteria.getKriteriaValue()));

    karakterKriteria.setKriteriaInputValue(calculateRangeVariableInpu
t(karakterKriteria.getRangeKriteriaInputValue(),
karakterKriteria.getKriteriaValue()));

    Iterator<InputRange> iteratorJaminanInputRange =
    jaminanKriteria.getKriteriaInputValue().keySet().iterator();
    List<ZN> listZn = new ArrayList<ZN>();
    while (iteratorJaminanInputRange.hasNext()) {

```

```

        InputRange jaminanInputRange =
        iteratorJaminanInputRange.next();
        Double jaminanInputValue =
        jaminanKriteria.getKriteriaInputValue().get(jaminanInputRange);

        Iterator<InputRange> iteratorUsahaInputRange =
        usahaKriteria.getKriteriaInputValue().keySet().iterator();
        while (iteratorUsahaInputRange.hasNext()) {
            InputRange usahaInputRange =
            iteratorUsahaInputRange.next();
            Double usahaInputValue =
            usahaKriteria.getKriteriaInputValue().get(usahaInputRange);

            Iterator<InputRange> iteratorKemampuanInputRange =
            kemampuanKriteria.getKriteriaInputValue().keySet().iterator();
            while (iteratorKemampuanInputRange.hasNext()) {
                InputRange kemampuanInputRange =
                iteratorKemampuanInputRange.next();
                Double kemampuanInputValue =
                kemampuanKriteria.getKriteriaInputValue().get(kemampuanInput
Range);

                Iterator<InputRange> iteratorKarakterInputRange =
                karakterKriteria.getKriteriaInputValue().keySet().iterator();
                while (iteratorKarakterInputRange.hasNext()) {
                    InputRange karakterInputRange =
                    iteratorKarakterInputRange.next();
                    Double karakterInputValue =
                    karakterKriteria.getKriteriaInputValue().get(karakterInputRange);

                    OutputRange outputRange =
                    getVariableOutput(jaminanInputRange, usahaInputRange,
kemampuanInputRange, karakterInputRange);
                    Kriteria kriteriaCalculate = new Kriteria();
                    kriteriaCalculate.setJaminan(jaminanInputValue);
                    kriteriaCalculate.setUsaha(usahaInputValue);

                    kriteriaCalculate.setKemampuan(kemampuanInputValue);
                    kriteriaCalculate.setKarakter(karakterInputValue);
                    ZN zn = new ZN();
                    zn.setA(kriteriaCalculate.getMinimumValue());

                    if(outputRange.equals(OutputRange.SANGAT_LAYAK)){
                        zn.setZ(getZValueIfSL(kriteriaCalculate.getMinimumValue()));
                    } else
                    if(outputRange.equals(OutputRange.LAYAK_DENGAN_RESIK
O_RENDAH)){
                        zn.setZ(getZValueIfLRR(kriteriaCalculate.getMinimumValue()));
                    } else
                    if(outputRange.equals(OutputRange.LAYAK_DENGAN_RESIK
O_TINGGI)){
                        zn.setZ(getZValueIfLRT(kriteriaCalculate.getMinimumValue()));
                    } else
                    if(outputRange.equals(OutputRange.KURANG_LAYAK)){
                        zn.setZ(getZValueIfKL(kriteriaCalculate.getMinimumValue()));
                    } else
                    if(outputRange.equals(OutputRange.SANGAT_TIDAK_LAYAK)){
                        zn.setZ(getZValueIfSTL(kriteriaCalculate.getMinimumValue()));
                    } else{
                        zn.setZ(0.0);
                    }
                    System.out.println("a ->\t" + zn.getA() + "\t-\t z->\t"
+ zn.getZ());
                    listZn.add(zn);
                }
            }
        }
    }
}

return calculateBigZ(listZn);

```

```

    }

public static void main(String[] args) {
    try {
        InputStream inputStream = new
FileInputStream("config.properties");
        Properties properties = new Properties();
        properties.load(inputStream);
        GlobVar.properties = properties;
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "File Config Not
Found.", "Error", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
    FuzzyLogic fuzzy = new FuzzyLogic();

//    Jaminan -> 10
//    Usaha -> 6.5
//    Kemampuan -> 4.5
//    Karakter -> 6.25
    Kriteria initKriteria = new Kriteria();
    initKriteria.setJaminan(10);
    initKriteria.setUsaha(6.5);
    initKriteria.setKemampuan(4.5);
    initKriteria.setKarakter(6.25);

    System.out.println(fuzzy.checkRangeVariableOutput(fuzzy.calcula
te(initKriteria)));
}
}

9. Kriteria.java

package org.uin.spk.logic;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

/**
 *
 * @author Haris_alkwitang
 */
public class Kriteria {

    private double jaminan;
    private double usaha;
    private double kemampuan;
    private double karakter;

    /**
     * @return the jaminan
     */
    public double getJaminan() {
        return jaminan;
    }

    /**
     * @param jaminan the jaminan to set
     */
    public void setJaminan(double jaminan) {
        this.jaminan = jaminan;
    }

    /**
     * @return the usaha
     */
    public double getUsaha() {
        return usaha;
    }

    /**
     * @param usaha the usaha to set
     */
    public void setUsaha(double usaha) {
        this.usaha = usaha;
    }

    /**
     * @return the kemampuan
     */
    public double getKemampuan() {
        return kemampuan;
    }

    /**
     * @param kemampuan the kemampuan to set
     */
    public void setKemampuan(double kemampuan) {
        this.kemampuan = kemampuan;
    }

    /**
     * @return the karakter
     */
    public double getKarakter() {
        return karakter;
    }

    /**
     * @param karakter the karakter to set
     */
    public void setKarakter(double karakter) {
        this.karakter = karakter;
    }

    public double getMinimumValue(){
        List<Double> listDouble = new ArrayList<Double>();
        listDouble.add(Double.valueOf(jaminan));
        listDouble.add(Double.valueOf(usaha));
        listDouble.add(Double.valueOf(kemampuan));
        listDouble.add(Double.valueOf(karakter));
        Collections.sort(listDouble);
        return listDouble.get(0);
    }
}

10. SingleKriteria.java

package org.uin.spk.logic;

import java.util.HashMap;
import java.util.List;

/**
 * @author haris_alkwitang
 */
public class SingleKriteria {

    public static enum KriteriaType{
        JAMINAN, USAHA, KEMAMPUAN, KARAKTER
    }
    private double kriteriaValue;
    private List<FuzzyLogic.InputRange> rangeKriteriaInputValue;
    private HashMap<FuzzyLogic.InputRange, Double>
kriteriaInputValue;

    /**
     * @return the kriteriaValue
     */
    public double getKriteriaValue() {
        return kriteriaValue;
    }

    /**
     * @param kriteriaValue the kriteriaValue to set
     */
    public void setKriteriaValue(double kriteriaValue) {
        this.kriteriaValue = kriteriaValue;
    }
}

```

```

    /**
     * @return the rangeKriteriaInputValue
     */
    public List<FuzzyLogic.InputRange>
    getRangeKriteriaInputValue() {
        return rangeKriteriaInputValue;
    }

    /**
     * @param rangeKriteriaInputValue the
     * rangeKriteriaInputValue to set
     */
    public void
    setRangeKriteriaInputValue(List<FuzzyLogic.InputRange>
    rangeKriteriaInputValue) {
        this.rangeKriteriaInputValue = rangeKriteriaInputValue;
    }

    /**
     * @return the kriteriaInputValue
     */
    public HashMap<FuzzyLogic.InputRange, Double>
    getKriteriaInputValue() {
        return kriteriaInputValue;
    }

    /**
     * @param kriteriaInputValue the kriteriaInputValue to set
     */
    public void
    setKriteriaInputValue(HashMap<FuzzyLogic.InputRange,
    Double> kriteriaInputValue) {
        this.kriteriaInputValue = kriteriaInputValue;
    }
}

```

## 11. ZN.java

```

package org.uin.spk.logic;

/**
 * * * @author haris_alkwitang
 */

public class ZN {
    private double a;
    private double z;

    /**
     * @return the a
     */
    public double getA() {
        return a;
    }

    /**
     * @param a the a to set
     */
    public void setA(double a) {
        this.a = a;
    }

    /**
     * @return the z
     */
    public double getZ() {
        return z;
    }

    /**
     * @param z the z to set
     */
    public void setZ(double z) {
        this.z = z;
    }
}

```

```

    /**
     * @return a * z ( a multiply by z)
     */
    public double getAxZ() {
        return a * z;
    }
}

```

## 12. Main.java

```

package org.uin.spk.main;

import org.uin.spk.screen.SplashFrame;
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
import javax.persistence.Persistence;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 * * * @author haris_alkwitang
 */

public class Main {

    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel("com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel");
        } catch (Exception e) {
            try {
                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
            } catch (Exception ee) {
            }
        }
        try {
            InputStream inputStream = new FileInputStream("config.properties");
            Properties properties = new Properties();
            properties.load(inputStream);
            GlobVar.properties = properties;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "File Config Not Found.", "Error", JOptionPane.ERROR_MESSAGE);
            System.exit(0);
        }
        try {
            Map<String, String> map = new HashMap<String, String>();
            map.put("javax.persistence.jdbc.url",
                    Function.getResources("db.url").replace("{ip}", Function.getResources("db.ipdb")).replace("{port}", Function.getResources("db.dbport")).replace("{dbname}", Function.getResources("db.dbname")));
            map.put("javax.persistence.jdbc.driver",
                    Function.getResources("db.driver").map.put("javax.persistence.jdbc.user",
                    Function.getResources("db.user")));
            map.put("javax.persistence.jdbc.password",
                    Function.getResources("db.pass"));
            GlobVar.entityManagerFactory =
            Persistence.createEntityManagerFactory("SPK-Pembiayaan_UKMPU", map);
            GlobVar.entityManagerFactory.createEntityManager().createQuery("Select l From Login l").setMaxResults(1).getResultList();
        }
    }
}

```



```

.addComponent(jLabel4,
javax.swing.GroupLayout.DEFAULT_SIZE, 331,
Short.MAX_VALUE)
.addComponent(jLabel5,
javax.swing.GroupLayout.DEFAULT_SIZE, 331,
Short.MAX_VALUE)
.addComponent(jLabel6,
javax.swing.GroupLayout.DEFAULT_SIZE, 331,
Short.MAX_VALUE)
.addComponent(jLabel7,
javax.swing.GroupLayout.DEFAULT_SIZE, 331,
Short.MAX_VALUE)
);
jPanel3Layout.setVerticalGroup

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel3Layout.createSequentialGroup()
.addContainerGap()
.addComponent(jLabel3)
.addGap(28, 28, 28)
.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabel7)
.addContainerGap(54, Short.MAX_VALUE))
);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(jLabel2,
javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
.addContainerGap()
.addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()
.addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()
.addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap())
);

.layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap()
);

layout.setVerticalGroup

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()

```

// Variables declaration - do not modify

```

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
// End of variables declaration
}

14. ImageDialog.java
/*
 * ImageDialog.java
 *
 * Created on Feb 19, 2011, 10:43:10 AM
 */
package org.uin.spk.screen;

import org.uin.spk.util.Function;

/**
 * * @author haris_alkwitang
 */

public class ImageDialog extends javax.swing.JDialog {

    public ImageDialog(java.awt.Frame parent, boolean modal,
byte[] byteImage) {
        super(parent, modal);
        initComponents();
        Function.setIcon(byteImage, jLabel1);
    }

    public ImageDialog(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
     * method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">
    private void initComponents() {

```

```

jScrollPane1 = new javax.swing.JScrollPane();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jScrollPane1.setViewportView(jLabel1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.DEFAULT_SIZE, 198, Short.MAX_VALUE)
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.DEFAULT_SIZE, 250, Short.MAX_VALUE)
    );
pack();
}// </editor-fold>

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}

15. KriteriaPanel.java
/*
 * KriteriaPanel.java
 * * * Created on Feb 14, 2011, 11:27:19 AM
 */
package org.uin.spk.screen;

import java.beans.Beans;
import java.util.ArrayList;
import java.util.List;
import javax.persistence.RollbackException;
import javax.swing.JPanel;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author haris_alkwitang
 */
public class KriteriaPanel extends JPanel {

    public KriteriaPanel() {
        initComponents();
        if (!Beans.isDesignTime()) {
            entityManager.getTransaction().begin();
        }
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
     * method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">

```

```

private void initComponents() {
    bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

    entityManager =
        GlobVar.entityManagerFactory.createEntityManager();
    query = java.beans.Beans.isDesignTime() ? null :
        entityManager.createQuery("SELECT k FROM KriteriaVariabel k
        ORDER BY k.id ASC");
    list = java.beans.Beans.isDesignTime() ?
        java.util.Collections.emptyList() :
        org.jdesktop.observablecollections.ObservableCollections.observableList(query.getResultList());
    masterScrollPane = new javax.swing.JScrollPane();
    masterTable = new javax.swing.JTable();
    variabelLabel = new javax.swing.JLabel();
    deskripsiLabel = new javax.swing.JLabel();
    variabelField = new javax.swing.JTextField();
    deskripsiField = new javax.swing.JTextField();
    saveButton = new javax.swing.JButton();
    refreshButton = new javax.swing.JButton();
    newButton = new javax.swing.JButton();
    deleteButton = new javax.swing.JButton();

    FormListener formListener = new FormListener();

    org.jdesktop.swingbinding.JTableBinding jTableBinding =
        org.jdesktop.swingbinding.SwingBindings.createTableBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE,
        list, masterTable);
    org.jdesktop.swingbinding.JTableBinding.ColumnBinding columnBinding =
        jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${id}"));
    columnBinding.setColumnName("Id");
    columnBinding.setColumnClass(String.class);
    columnBinding =
        jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${variabel}"));
    columnBinding.setColumnName("Variabel");
    columnBinding.setColumnClass(String.class);
    columnBinding =
        jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${deskripsi}"));
    columnBinding.setColumnName("Deskripsi");
    columnBinding.setColumnClass(String.class);
    bindingGroup.addBinding(jTableBinding);
    jTableBinding.bind();
    masterScrollPane.setViewportView(masterTable);
    variabelLabel.setText("Variabel:");
    deskripsiLabel.setText("Deskripsi:");

    org.jdesktop.beansbinding.Binding binding =
        org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE,
        masterTable,
        org.jdesktop.beansbinding.ELProperty.create("${selectedElement.variabel}"),
        variabelField,
        org.jdesktop.beansbinding.BeanProperty.create("text"));
    binding.setSourceUnreadableValue(null);
    bindingGroup.addBinding(binding);
    binding =
        org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ,
        masterTable,
        org.jdesktop.beansbinding.ELProperty.create("${selectedElement!= null}"),
        variabelField,
        org.jdesktop.beansbinding.BeanProperty.create("enabled"));
    bindingGroup.addBinding(binding);
    binding =
        org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE,
        masterTable,
        org.jdesktop.beansbinding.ELProperty.create("${selectedElement.deskripsi}"),
        deskripsiField,
        org.jdesktop.beansbinding.BeanProperty.create("text"));

```



```

        KriteriaPanel.this.deleteButtonActionPerformed(evt);
    }
}
//</editor-fold>

@SuppressWarnings("unchecked")
private void
refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
    entityManager.getTransaction().rollback();
    entityManager.getTransaction().begin();
    java.util.Collection data = query.getResultList();
    for (Object entity : data) {
        entityManager.refresh(entity);
    }
    list.clear();
    list.addAll(data);
}

private void
deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
    int[] selected = masterTable.getSelectedRows();
    List<org.uin.spk.entity.KriteriaVariabel> toRemove = new
    ArrayList<org.uin.spk.entity.KriteriaVariabel>(selected.length);
    for (int idx=0; idx<selected.length; idx++) {
        org.uin.spk.entity.KriteriaVariabel k =
    list.get(masterTable.convertRowIndexToModel(selected[idx]));
        toRemove.add(k);
        entityManager.remove(k);
    }
    list.removeAll(toRemove);
}

private void
newButtonActionPerformed(java.awt.event.ActionEvent evt) {
    org.uin.spk.entity.KriteriaVariabel k = new
    org.uin.spk.entity.KriteriaVariabel(Function.getPKKriteriaVariabel
    ());
    entityManager.persist(k);
    list.add(k);
    int row = list.size()-1;
    masterTable.setRowSelectionInterval(row, row);

    masterTable.scrollRectToVisible(masterTable.getCellRect(row, 0,
    true));
}

private void
saveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        entityManager.getTransaction().commit();
        entityManager.getTransaction().begin();
    } catch (RollbackException rex) {
        rex.printStackTrace();
        entityManager.getTransaction().begin();
        List<org.uin.spk.entity.KriteriaVariabel> merged = new
        ArrayList<org.uin.spk.entity.KriteriaVariabel>(list.size());
        for (org.uin.spk.entity.KriteriaVariabel k : list) {
            merged.add(entityManager.merge(k));
        }
        list.clear();
        list.addAll(merged);
    }
}

// Variables declaration - do not modify
private javax.swing.JButton deleteButton;
private javax.swing.JTextField deskripsiField;
private javax.swing.JLabel deskripsiLabel;
private javax.persistence.EntityManager entityManager;
private java.util.List<org.uin.spk.entity.KriteriaVariabel> list;
private javax.swing.JScrollPane masterScrollPane;
private javax.swing.JTable masterTable;
private javax.swing.JButton newButton;
private javax.persistence.Query query;
private javax.swing.JButton refreshButton;
private javax.swing.JButton saveButton;

```

```

private javax.swing.JTextField variabelField;
private javax.swing.JLabel variabelLabel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration
}

```

## 16. LaporanPanel.java

```

/*
 * LaporanPanel.java
 *
 * Created on Feb 20, 2011, 3:41:46 PM
 */
package org.uin.spk.screen;

import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import javax.persistence.EntityManager;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import org.uin.spk.entity.MasterPerhitungan;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.Perhitungan;
import org.uin.spk.entity.controller.MasterPerhitunganJpaController;
import org.uin.spk.entity.controller.PemohonJpaController;
import org.uin.spk.entity.controller.PerhitunganJpaController;
import org.uin.spk.logic.FuzzyLogic;
import org.uin.spk.logic.Kriteria;
import org.uin.spk.util.Function;

/**
 *
 * @author haris_alkwitang
 */
public class LaporanPanel extends javax.swing.JPanel {

    PemohonJpaController pemohonController = new
    PemohonJpaController();
    MasterPerhitunganJpaController masterPerhitunganController =
    new MasterPerhitunganJpaController();
    PerhitunganJpaController perhitunganController = new
    PerhitunganJpaController();
    EntityManager entityManager = Function.getEntityManager();
    HashMap<Integer, String> mapPK = new HashMap<Integer,
    String>();

    /** Creates new form LaporanPanel */
    public LaporanPanel() {
        initComponents();
        refresh();
        setLabel();
    }

    private void setLabel() {
        //
        jLabel4.setText(Function.getResources("label.namapemohon"));
        //
        jLabel5.setText(Function.getResources("label.nilaipinjaman"));
        //
        jLabel6.setText(Function.getResources("label.rekomendasi2"));
        // jLabel7.setText(Function.getResources("label.hasilfuzzy"));
        // jButton1.setText(Function.getResources("button.delete"));
    }

    private void setHeaderTable() {
        DefaultTableModel tableModel = new DefaultTableModel();

        tableModel.addColumn(Function.getResources("table.header.nope
        mohon"));

        tableModel.addColumn(Function.getResources("table.header.jenis
        pemohon"));
    }
}

```

```

tableModel.addColumn(Function.getResources("table.header.nama
pemohon"));

tableModel.addColumn(Function.getResources("table.header.nilai
pinjaman"));

tableModel.addColumn(Function.getResources("table.header.nilai
hasil"));

tableModel.addColumn(Function.getResources("table.header.range
hasil"));
    jTable1.setModel(tableModel);
}

private void refresh() {
    setHeaderTable();
    DefaultTableModel dtm = (DefaultTableModel)
jTable1.getModel();
    List<MasterPerhitungan> listPerhitungan =
Function.getEntityManager().createNamedQuery("MasterPerhitun
gan.findAll").getResultSet();
    mapPK = new HashMap<Integer, String>();
    int i = 0;
    for (MasterPerhitungan masterPerhitungan : listPerhitungan) {
        Pemohon pemohon =
pemohonController.findPemohon(masterPerhitungan.getNoCalon
Debitur);
        List returnList = calculate(masterPerhitungan);
        // NumberFormat nf = new
DecimalFormat(Function.getResources("application.numberformat
"));
        NumberFormat nf = new DecimalFormat("#,###.00");
        try {
            dtm.addRow(new
Object[]{pemohon.getNoCalonDebitur(),
pemohon.getJenisPemohon(), pemohon.getNamaCalonDebitur(),
nf.format(Double.valueOf(pemohon.getNilaiJaminan())),
Double.valueOf(returnList.get(0).toString()),
String.valueOf(returnList.get(1)))});
            mapPK.put(i,
masterPerhitungan.getIdMasterPerhitungan());
            i++;
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    // jProgressBar1.setValue(0);
    // jTextField1.setText("");
    // jTextField2.setText("");
    // jFormattedTextField1.setText("");
    // jLabel1.setText("");
}

private List calculate(MasterPerhitungan masterPerhitungan) {
    List returnList = new ArrayList();
    FuzzyLogic fuzzyLogic = new FuzzyLogic();
    double fuzzyValue =
fuzzyLogic.calculate(doFill(masterPerhitungan));
    returnList.add(fuzzyValue);
    FuzzyLogic.OutputRange outputRange =
fuzzyLogic.checkRangeVariableOutput(fuzzyValue);
    if
(outputRange.equals(FuzzyLogic.OutputRange.SANGAT_LAYA
K)) {
        returnList.add(Function.getResources("system.sl"));
    } else if
(outputRange.equals(FuzzyLogic.OutputRange.LAYAK_DENGA
N_RESIKO_RENDAH)) {
        returnList.add(Function.getResources("system.lrr"));
    } else if
(outputRange.equals(FuzzyLogic.OutputRange.LAYAK_DENGA
N_RESIKO_TINGGI)) {
        returnList.add(Function.getResources("system.lrt"));
    }
}

    } else if
(outputRange.equals(FuzzyLogic.OutputRange.KURANG_LAYA
K)) {
    returnList.add(Function.getResources("system.kl"));
} else if
(outputRange.equals(FuzzyLogic.OutputRange.SANGAT_TIDAK
_LAYAK)) {
    returnList.add(Function.getResources("system.stl"));
} else {
    returnList.add("Error");
}
return returnList;
}

private Kriteria doFill(MasterPerhitungan masterPerhitungan) {
    Kriteria kriteria = new Kriteria();
    kriteria.setJaminan(0.0);
    kriteria.setKarakter(0.0);
    kriteria.setKemampuan(0.0);
    kriteria.setUsaha(0.0);
    List<Perhitungan> listPerhitungan =
entityManager.createNamedQuery("Perhitungan.findByIdMasterP
erhitungan").setParameter("idMasterPerhitungan",
masterPerhitungan.getIdMasterPerhitungan()).getResultSet();
    for (Perhitungan perhitungan : listPerhitungan) {
        if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.nilaijaminan"))) {
            kriteria.setJaminan(Double.parseDouble(perhitungan.getNilai() +
 ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.penghasilan"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.pengeluaran"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.kepemilikantempattinggal"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.kondisitempattinggal"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.modalinvestasi"))) {
            kriteria.setUsaha(kriteria.getUsaha() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.jumlahtanggungan"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.kondisitempattinggal"))) {
            kriteria.setKemampuan(kriteria.getKemampuan() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.usaha"))) {
            kriteria.setUsaha(kriteria.getUsaha() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.lokasiusaha"))) {
            kriteria.setUsaha(kriteria.getUsaha() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        } else if
(perhitungan.getPerhitunganPK().getIdKriteria().equals(Function.g
etResources("key.child.jenistempatusaha"))) {
            kriteria.setUsaha(kriteria.getUsaha() +
 Double.parseDouble(perhitungan.getNilai() + ""));
        }
    }
}

```

```
        } else if
(perhitungan.getIdKriteria().equals(Function.get
etResources("key.child.lamausaha"))) {
        kriteria.setUsaha(kriteria.getUsaha() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.persaingan"))) {
        kriteria.setUsaha(kriteria.getUsaha() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.kepemilikantempatusaha"))) {
        kriteria.setUsaha(kriteria.getUsaha() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.jumlahtenagakerja"))) {
        kriteria.setUsaha(kriteria.getUsaha() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.perilaku"))) {
        kriteria.setKarakter(kriteria.getKarakter() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.penampilan"))) {
        kriteria.setKarakter(kriteria.getKarakter() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.interaksi"))) {
        kriteria.setKarakter(kriteria.getKarakter() +
Double.parseDouble(perhitungan.getNilai() + ""));
    } else if
(perhitungan.getIdKriteria().equals(Function.g
etResources("key.child.ketaatan"))) {
        kriteria.setKarakter(kriteria.getKarakter() +
Double.parseDouble(perhitungan.getNilai() + ""));
    }
}
return kriteria;
}

private void doBinding() {
    String selectedValue =
jTable1.getValueAt(jTable1.getSelectedRow(), 4).toString();
    Double value = Double.parseDouble(selectedValue);
    //
    jProgressBar1.setValue(value.intValue());
    //
    jLabel1.setText(selectedValue);
    //
jTextField1.setText(jTable1.getValueAt(jTable1.getSelectedRow()
, 2).toString());
    //
jTextField2.setText(jTable1.getValueAt(jTable1.getSelectedRow()
, 5).toString());
    //
jFormattedTextField1.setValue(Double.valueOf(jTable1.getValue
At(jTable1.getSelectedRow(),
3).toString().split("[.]")[0].replace(",", "")));
}

private void doDelete() {
    if (jTable1.getSelectedRow() != -1) {
        if (JOptionPane.showConfirmDialog(null,
Function.getResources("alert.delete.perhitungan"), "Alert",
JOptionPane.YES_NO_OPTION) ==
JOptionPane.YES_OPTION) {
            if (mapPK.containsKey(jTable1.getSelectedRow())) {
                try {
masterPerhitunganController.destroy(mapPK.get(jTable1.getSelect
edRow()));
                } catch (Exception e) {
e.printStackTrace();

```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(jButton1)
    .addContainerGap()
);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addGap(10, 10, 10))
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 671,
Short.MAX_VALUE)
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 493,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGapContainerGap(71, Short.MAX_VALUE))
);
}//</editor-fold>

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    doBinding();
}

private void jTable1KeyReleased(java.awt.event.KeyEvent evt) {
    doBinding();
}

private void
 jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    doDelete();
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

## 17. LoginFrame.java

```

/*
 * LoginFrame.java
 *
 * * Created on Feb 14, 2011, 8:59:07 AM
 */
package org.uin.spk.screen;

```

```

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import org.uin.spk.entity.Login;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 * * * @author haris_alkwitang
 */
public class LoginFrame extends javax.swing.JFrame {

    /* Creates new form LoginFrame */
    public LoginFrame() {
        initComponents();
        setLabel();
        if
        (!Function.getResources("application.picture.login").equals(""))
        Function.setIcon(Function.getResources("application.picture.login"),
        jLabel4);
    }

    private void setLabel() {
        setTitle(Function.getResources("application.login.title"));
        jButton1.setText(Function.getResources("button.login"));
        jButton2.setText(Function.getResources("button.cancel"));
        jLabel1.setText(Function.getResources("label.login.header"));

        jLabel6.setText(Function.getResources("label.login2.header"));
        jLabel2.setText(Function.getResources("label.username"));
        jLabel3.setText(Function.getResources("label.password"));
    }

    private void doLogin() {
        EntityManager entityManager =
        GlobVar.entityManagerFactory.createEntityManager();
        Query q = entityManager.createQuery("Select l From Login l
Where l.username = :uname and l.password = :pass");
        q.setParameter("uname", jTextField1.getText());
        q.setParameter("pass", jPasswordField1.getText());
        List l = q.getResultList();
        if (l != null && !l.isEmpty()) {
            Login login = (Login) l.get(0);
            MainFrame mainFrame = new MainFrame(login);
            mainFrame.setExtendedState(JFrame.MAXIMIZED_BOTH);
            mainFrame.setVisible(true);
            GlobVar.mainFrame = mainFrame;
            this.dispose();
        } else {
            JOptionPane.showMessageDialog(null,
            Function.getResources("alert.login.failedlogin"));
        }
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
     * method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">
    private void initComponents() {

        jLabel5 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();

```



```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jButton2)
    .addComponent(jButton1))
.addContainerGap()
);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
);
);

pack();
}// </editor-fold>

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    doLogin();
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;

```

```

private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

### 18. MainFrame.java

```

/*
 * MainFrame.java
 *
 * Created on Feb 14, 2011, 8:52:05 AM
 */
package org.uin.spk.screen;

import javax.swing.JPanel;
import javax.swing.JScrollPane;
import org.uin.spk.entity.Login;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author haris_alkwitang
 */
public class MainFrame extends javax.swing.JFrame {

    Login loginUser;

    /**
     * Creates new form MainFrame
     */
    public MainFrame(Login login) {
        initComponents();
        setLabel();
        loginUser = login;
        GlobVar.loginUser = loginUser;
        setACL();
        setPanel(new MainMenuPanel());
    }

    private void setACL() {
        if
(loginUser.getRule().equalsIgnoreCase(Function.getResources("ap
lication.admin")))
{
    setRule("admin");
} else if
(loginUser.getRule().equalsIgnoreCase(Function.getResources("ap
lication.manager")))
{
    setRule("manager");
}
}

private void setRule(String rule) {
    if (!Function.getResources("application." + rule +
".kriteria").equalsIgnoreCase("1"))
        && !Function.getResources("application." + rule +
".nilaikriteria").equalsIgnoreCase("1")) {
        jMenuItem2.setVisible(false);
} else {
    if (Function.getResources("application." + rule +
".nilaikriteria").equalsIgnoreCase("1")) {
        jMenuItem2.setVisible(true);
} else {
        jMenuItem2.setVisible(false);
}
if (Function.getResources("application." + rule +
".kriteria").equalsIgnoreCase("1")) {
        jMenuItem1.setVisible(true);
} else {
        jMenuItem1.setVisible(false);
}
}
}

```

```
        }

        if (Function.getResources("application." + rule +
".pemohon").equalsIgnoreCase("1")) {
            jMenuItem3.setVisible(true);
        } else {
            jMenuItem3.setVisible(false);
            jMenuItem3.setLabel("Pemohon");
        }

        if (!Function.getResources("application." + rule +
".pemeriksaan").equalsIgnoreCase("1") ||
&& !Function.getResources("application." + rule +
".perhitungan").equalsIgnoreCase("1")) {
            jMenuItem4.setVisible(false);
        } else {
            if (Function.getResources("application." + rule +
".pemeriksaan").equalsIgnoreCase("1")) {
                jMenuItem4.setVisible(true);
            } else {
                jMenuItem5.setVisible(false);
            }
            if (Function.getResources("application." + rule +
".perhitungan").equalsIgnoreCase("1")) {
                jMenuItem4.setVisible(true);
            } else {
                jMenuItem5.setVisible(false);
            }
        }
    }

    if (Function.getResources("application." + rule +
".laporan").equalsIgnoreCase("1")) {
        jMenuItem9.setVisible(true);
    } else {
        jMenuItem9.setVisible(false);
        jMenuItem6.setVisible(false);
    }
}

private void setLabel() {
    setTitle(Function.getResources("application.title"));
    jMenuItem1.setText(Function.getResources("menu.file"));
    jMenuItem2.setText(Function.getResources("menu.kriteria"));
    jMenuItem3.setText(Function.getResources("menu.pemohon"));

jMenuItem4.setText(Function.getResources("menu.pemeriksaan"));
    jMenuItem5.setText(Function.getResources("menu.about"));
    jMenuItem6.setText(Function.getResources("menu.laporan"));

jMenuItem1.setText(Function.getResources("menu.item.kriteria"));
;
jMenuItem2.setText(Function.getResources("menu.item.nilaikritia"));
jMenuItem3.setText(Function.getResources("menu.item.pemohon"));
);
jMenuItem4.setText(Function.getResources("menu.item.pemeriksaan"));
jMenuItem5.setText(Function.getResources("menu.item.perhitungan"));
jMenuItem7.setText(Function.getResources("menu.item.mainmenu"));
jMenuItem6.setText(Function.getResources("menu.item.exit"));
jMenuItem8.setText(Function.getResources("menu.item.about"));
jMenuItem9.setText(Function.getResources("menu.item.laporan"));
;

}

public static void setPanel(JPanel panel) {
    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setViewportView(panel);
    mainPanel.removeAll();
    mainPanel.add(scrollPane);
    mainPanel.revalidate();
    mainPanel.repaint();
}
```

```
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    jPanel1 = new javax.swing.JPanel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem7 = new javax.swing.JMenuItem();
    jMenuItem6 = new javax.swing.JMenuItem();
    jMenu2 = new javax.swing.JMenu();
    jMenuItem2 = new javax.swing.JMenuItem();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenu3 = new javax.swing.JMenu();
    jMenuItem3 = new javax.swing.JMenuItem();
    jMenu4 = new javax.swing.JMenu();
    jMenuItem4 = new javax.swing.JMenuItem();
    jMenuItem5 = new javax.swing.JMenuItem();
    jMenu6 = new javax.swing.JMenu();
    jMenuItem9 = new javax.swing.JMenuItem();
    jMenu5 = new javax.swing.JMenu();
    jMenuItem8 = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_O
N_CLOSE);
    setResizable(false);
    mainPanel.setLayout(new java.awt.BorderLayout());
    jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder(
));
    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
)
        .createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
)
        .createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 22, Short.MAX_VALUE)
    );
    jMenu1.setText("File");
    jMenuItem7.setText("Main Page");
    jMenuItem7.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jMenuItem7ActionPerformed(evt);
        }
    });
    jMenu1.add(jMenuItem7);

    jMenuItem6.setText("Exit");
    jMenuItem6.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jMenuItem6ActionPerformed(evt);
        }
    });
    jMenu1.add(jMenuItem6);

    jMenuBar1.add(jMenu1);
}
```

```

jMenu2.setText(" Kriteria");
jMenuItem2.setText("Nilai Kriteria");
jMenuItem2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuItem1.setText("Kriteria");
jMenuItem1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem1);

jMenuBar1.add(jMenu2);

jMenu3.setText(" Pemohon");
jMenuItem3.setText("Pemohon");
jMenuItem3.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem3ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem3);

jMenuBar1.add(jMenu3);

jMenu4.setText(" Pemeriksaan");
jMenuItem4.setText("Pemeriksaan");
jMenuItem4.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem4ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem4);

jMenuItem5.setText("Perhitungan");
jMenuItem5.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem5ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem5);

jMenuBar1.add(jMenu4);

jMenu6.setText(" Laporan");
jMenuItem9.setText(" Laporan");
jMenuItem9.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem9ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem9);

jMenuBar1.add(jMenu6);

jMenu5.setText("About");
jMenuItem8.setText(" About");
jMenuItem8.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jMenuItem8ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem8);

jMenuBar1.add(jMenu5);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addComponent(mainPanel,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 844,
                    Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jPanel1,
                    javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addComponent(mainPanel,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 520,
                    Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jPanel1,
                    javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
);

pack();
}// </editor-fold>

private void
jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new KriteriaPanel());
}
private void
jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new NilaiKriteriaPanel());
}
private void
jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new PemohonPanel());
}
private void
jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new PemeriksaanPanel());
}
private void
jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new PerhitunganPanel());
}
private void
jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

```

```

}

private void
jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new MainMenuPanel());
}

private void
jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
    AboutDialog aboutDialog = new AboutDialog(this, true);
    aboutDialog.setLocationRelativeTo(null);
    aboutDialog.setVisible(true);
}

private void
jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
    setPanel(new LaporanPanel());
}

// Variables declaration - do not modify
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenu jMenu4;
private javax.swing.JMenu jMenu5;
private javax.swing.JMenu jMenu6;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem5;
private javax.swing.JMenuItem jMenuItem6;
private javax.swing.JMenuItem jMenuItem7;
private javax.swing.JMenuItem jMenuItem8;
private javax.swing.JMenuItem jMenuItem9;
private javax.swing.JPanel jPanel1;
private static javax.swing.JPanel mainPanel;
// End of variables declaration
}

19. MainMenuPanel.java
/*
 * MainMenuPanel.java
 *
 * Created on Feb 20, 2011, 5:10:03 PM
 */
package org.uin.spk.screen;

import java.net.URL;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author haris_alkwitang
 */
public class MainMenuPanel extends javax.swing.JPanel {

    /** Creates new form MainMenuPanel */
    public MainMenuPanel() {
        initComponents();
        String url = "";
        if
(GlobVar.loginUser.getRule().equalsIgnoreCase(Function.getResources("application.admin"))){
            url="file:/// +
Function.getResources("file.mainmenu.admin.absolutepath");
        } else {
            url="file:/// +
Function.getResources("file.mainmenu.manager.absolutepath");
        }
        try {
            jEditorPane1.setPage(new URL(url));
        } catch (Exception e) {
}
}

}

}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jEditorPane1 = new javax.swing.JEditorPane();

    jEditorPane1.setEditable(false);
    jScrollPane1.setViewportView(jEditorPane1);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
    this.setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 400,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 300,
Short.MAX_VALUE)
    );
// </editor-fold>

// Variables declaration - do not modify
private javax.swing.JEditorPane jEditorPane1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}

20. MasterPerhitunganDialog.java
/*
 * PemohonDialog.java
 *
 * Created on Feb 15, 2011, 6:54:50 PM
 */
package org.uin.spk.screen;

import java.util.List;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import org.uin.spk.entity.MasterPerhitungan;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.controller.MasterPerhitunganJpaController;
import org.uin.spk.entity.controller.PemohonJpaController;
import org.uin.spk.entity.controller.PerhitunganJpaController;
import org.uin.spk.util.Function;

/**
 *
 * @author haris_alkwitang
 */
public class MasterPerhitunganDialog extends javax.swing.JDialog {

    PemohonJpaController pemohonController = new
PemohonJpaController();
}

```

```

MasterPerhitunganJpaController masterPerhitunganController =
new MasterPerhitunganJpaController();
PerhitunganJpaController perhitunganController = new
PerhitunganJpaController();
Pemohon pemohonSelected;
MasterPerhitungan masterPerhitunganSelected;

/** Creates new form PemohonDialog */
public MasterPerhitunganDialog(java.awt.Frame parent, boolean
modal) {
    super(parent, modal);
    initComponents();
    refresh();
}

private void setHeaderTable() {
    DefaultTableModel tableModel = new DefaultTableModel();

    tableModel.addColumn(Function.getResources("table.header.idper
hitungan"));

    tableModel.addColumn(Function.getResources("table.header.nope
mohon"));

    tableModel.addColumn(Function.getResources("table.header.nama
pemohon"));

    tableModel.addColumn(Function.getResources("table.header.dateh
itung"));
    jTable1.setModel(tableModel);
}

private void refresh() {
    setHeaderTable();
    DefaultTableModel dtm = (DefaultTableModel)
jTable1.getModel();
    List<MasterPerhitungan> listPerhitungan =
masterPerhitunganController.findMasterPerhitunganEntities();
    for (MasterPerhitungan masterPerhitungan : listPerhitungan) {
        Pemohon pemohon =
pemohonController.findPemohon(masterPerhitungan.getNoCalon
Debitur());
        dtm.addRow(new
Object[]{masterPerhitungan.getIdMasterPerhitungan(),
pemohon.getNoCalonDebitur(), pemohon.getNamaCalonDebitur(),
masterPerhitungan.getDateCheck()});
    }
}

public MasterPerhitungan getMasterPerhitunganSelected(){
    return masterPerhitunganSelected;
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new JTable(){
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    jScrollPane1.setViewportView(jTable1);

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOS
E_ON_CLOSE);

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            new Object [] {
                "", new String [] {
                    ""
                }
            }
        },
        new String [] {
            ""
        }
    ));
    jTable1.setToolTipText("Double Click to Choose Pemohon");
    jTable1.addMouseListener(new
java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt)
        {
            jTable1MouseClicked(evt);
        }
    });
    jScrollPane1.setViewportView(jTable1);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 471,
Short.MAX_VALUE)
        )
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 399,
Short.MAX_VALUE))
                .addComponent(layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(pack(), javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE))
            )
        )
    )
);
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 399,
Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(pack(), javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE))
        )
);
}
// Variables declaration - do not modify
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

21. NilaiKriteriaPanel.java
/*
 * NilaiKriteriaPanel.java
 *
 * Created on Feb 14, 2011, 12:26:24 PM
 */
package org.uin.spk.screen;

import java.util.HashMap;
import java.util.List;
import javax.swing.table.DefaultTableModel;
import javax.swing.*;
import org.uin.spk.entity.KriteriaNilai;
import org.uin.spk.entity.KriteriaVariabel;
import org.uin.spk.entity.controller.KriteriaNilaiJpaController;
import org.uin.spk.entity.controller.KriteriaVariabelJpaController;
import org.uin.spk.util.Function;

*/
*
* @author Haris_alkwitang
*/

```

```

public class NilaiKriteriaPanel extends javax.swing.JPanel {
    KriteriaNilaiJpaController kriteriaNilaiController = new
    KriteriaNilaiJpaController();
    KriteriaVariabelJpaController kriteriaVariableController = new
    KriteriaVariabelJpaController();
    HashMap mapVariable, mapVariable2;
    HashMap valueList, valueList2;

    /** Creates new form NilaiKriteriaPanel */
    public NilaiKriteriaPanel() {
        initComponents();
        loadCombo();
        setHeaderTable();
        refresh();
        setLabel();
    }

    private void loadCombo() {
        List<KriteriaVariabel> listVariable =
        kriteriaVariableController.findKriteriaVariabelEntities();
        valueList = new HashMap();
        mapVariable = new HashMap();
        for (KriteriaVariabel kriteriaVariable : listVariable) {
            valueList.put(kriteriaVariable.getVariabel(),
kriteriaVariable.getId());
            mapVariable.put(kriteriaVariable.getId().toString(),
kriteriaVariable.getVariabel());
        }
        zCombo1.setNilai(valueList);
    }

    List<KriteriaNilai> listVal =
    kriteriaNilaiController.findKriteriaNilaiEntities();
    mapVariable2 = new HashMap();
    mapVariable2.put("", "");
    for (KriteriaNilai kriteriaNilai : listVal) {
        mapVariable2.put(kriteriaNilai.getIdKriteria().toString(),
kriteriaNilai.getParameter());
    }
    loadSecondCombo();
}

private void loadSecondCombo() {
    List<KriteriaNilai> listVal =
    kriteriaNilaiController.findKriteriaNilaiEntities();
    valueList2 = new HashMap();
    valueList2.put("", "");
    for (KriteriaNilai kriteriaNilai : listVal) {
        if ((kriteriaNilai.getIdKriteriaInduk() == null ||
kriteriaNilai.getIdKriteriaInduk() == 0)
        &&
String.valueOf(kriteriaNilai.getIdVariabel()).equals(zCombo1.getElectedKey().toString())) {
            valueList2.put(kriteriaNilai.getParameter(),
kriteriaNilai.getIdKriteria());
        }
    }
    zCombo2.setNilai(valueList2);
}

private void setHeaderTable() {
    DefaultTableModel tableModel = new DefaultTableModel();

    tableModel.addColumn(function.getResources("table.header.idkrit
eria"));
    tableModel.addColumn(function.getResources("table.header.kriter
iavariable"));
    tableModel.addColumn(function.getResources("table.header.kriter
iainduk"));
    tableModel.addColumn(function.getResources("table.header.para
meter"));
    tableModel.addColumn(function.getResources("table.header.bobo
t"));

    masterTable.setModel(tableModel);
}
}

private void removeTableRows() {
    setHeaderTable();
    DefaultTableModel dtm = (DefaultTableModel)
masterTable.getModel();
    for (int i = 1; i <= dtm.getRowCount(); i++) {
        dtm.removeRow(0);
    }
}

private void setLabel() {
    saveButton.setText(function.getResources("button.save"));
    newButton.setText(function.getResources("button.new"));

    deleteButton.setText(function.getResources("button.delete"));
    refreshButton.setText(function.getResources("button.refresh"));
}

private void refresh() {
    removeTableRows();
    DefaultTableModel dtm = (DefaultTableModel)
masterTable.getModel();
    List<KriteriaNilai> listNilai =
    kriteriaNilaiController.findKriteriaNilaiEntities();
    for (KriteriaNilai kriteriaNilai : listNilai) {
        dtm.addRow(new Object[]{kriteriaNilai.getIdKriteria(),
mapVariable.get(kriteriaNilai.getIdVariabel() + ""),
kriteriaNilai.getIdKriteriaInduk() == null ||
kriteriaNilai.getIdKriteriaInduk() == 0 ? "" :
mapVariable2.get(kriteriaNilai.getIdKriteriaInduk() + ""),
kriteriaNilai.getParameter(), kriteriaNilai.getBobot()});
    }
    idKriteriaField.setText("");
    jFormattedTextField1.setValue(0.0);
    parameterField.setText("");
    zCombo1.setSelectedIndex(0);
    zCombo2.setSelectedIndex(0);
    loadCombo();
}

private void binding() {
    idKriteriaField.setText(masterTable.getValueAt(masterTable.getSe
lectedRow(), 0).toString());
    //
    zCombo2.setSelectedItem(masterTable.getValueAt(masterTable.g
etSelectedRow(), 2) == null ? "" :
masterTable.getValueAt(masterTable.getSelectedRow(),
2).toString());
    zCombo2.setSelectedItem(masterTable.getValueAt(masterTable.g
etSelectedRow(), 2).toString());
    zCombo1.setSelectedItem(masterTable.getValueAt(masterTable.g
etSelectedRow(), 1).toString());
    parameterField.setText(masterTable.getValueAt(masterTable.getSe
lectedRow(), 3).toString());

    jFormattedTextField1.setValue(Double.parseDouble(masterTable.
getValueAt(masterTable.getSelectedRow(), 4).toString()));
}

private boolean doCheck() {
    return function.checkIsi(new JComponent[]{zCombo1,
parameterField, jFormattedTextField1});
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {
    jScrollPane1 = new javax.swing.JScrollPane();
    masterTable = new JTable();
}

```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(parameterLabel)
        .addComponent(bobotLabel))
    .addGap(5, 5, 5)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(parameterField,
        javax.swing.GroupLayout.DEFAULT_SIZE, 466,
        Short.MAX_VALUE)
            .addComponent(jFormattedTextField1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 93,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap()

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addComponent(newButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(deleteButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(refreshButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(saveButton)
    .addContainerGap()))
);

layout.linkSize(javax.swing.SwingConstants.HORIZONTAL,
new java.awt.Component[] {bobotLabel, idKriteriaLabel,
idVariabelLabel1, idVariabelLabel2, parameterLabel});

layout.setVerticalGroup()

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane1,
        javax.swing.GroupLayout.DEFAULT_SIZE, 255,
        Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(idKriteriaLabel)
        .addComponent(idKriteriaField,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(idVariabelLabel2)
        .addComponent(zCombo1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(idVariabelLabel1)
        .addComponent(zCombo2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(parameterLabel)
        .addComponent(parameterField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(bobotLabel)
        .addComponent(jFormattedTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(saveButton)
        .addComponent(refreshButton)
        .addComponent(deleteButton)
        .addComponent(newButton))
        .addContainerGap()

});

}//</editor-fold>

private void
newButtonActionPerformed(java.awt.event.ActionEvent evt) {
    refresh();
}

private void
deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        kriteriaNilaiController.destroy(new
Integer(masterTable.getValueAt(masterTable.getSelectedRow(),
0).toString()));
    } catch (Exception e) {
        e.printStackTrace();
    }
    refresh();
}

private void
refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
    refresh();
}

private void
saveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if (doCheck()) {
        try {
            KriteriaNilai kriteriaNilai;
            if (idKriteriaField.getText().equals("")) {
                kriteriaNilai = new KriteriaNilai();
                kriteriaNilai.setIdVariabel(new
Integer(zCombo1.getSelectedItem().toString()).intValue());
                kriteriaNilai.setParameter(parameterField.getText());
            } else {
                kriteriaNilai = kriteriaNilaiController.get(Integer.parseInt(idKriteriaField.getText()));
                kriteriaNilai.setParameter(parameterField.getText());
            }
            kriteriaNilaiController.update(kriteriaNilai);
            refresh();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

kriteriaNilai.setBobot(Double.valueOf(jFormattedTextField1.getText().toString()));

kriteriaNilai.setIdKriteriaInduk(zCombo2.getSelectedKey().toString() == "" ? null : new Integer(zCombo2.getSelectedKey().toString().intValue()));
    kriteriaNilaiController.create(kriteriaNilai);
} else {
    kriteriaNilai =
kriteriaNilaiController.findKriteriaNilai(new Integer(masterTable.getValueAt(masterTable.getSelectedRow(), 0).toString()));
    kriteriaNilai.setIdVariabel(new Integer(zCombo1.getSelectedKey().toString().intValue()));
    kriteriaNilai.setParameter(parameterField.getText());

kriteriaNilai.setBobot(Double.parseDouble(jFormattedTextField1.getText().toString()));

kriteriaNilai.setIdKriteriaInduk(zCombo2.getSelectedKey().toString() == "" ? null : new Integer(zCombo2.getSelectedKey().toString().intValue()));
    kriteriaNilaiController.edit(kriteriaNilai);
}
refresh();
} catch (Exception e) {
    e.printStackTrace();
}
}

private void masterTableMouseClicked(java.awt.event.MouseEvent evt) {
    binding();
}

private void zCombo1ActionPerformed(java.awt.event.ActionEvent evt) {
    loadSecondCombo();
}

private void masterTableKeyReleased(java.awt.event.KeyEvent evt) {
    binding();
}

// Variables declaration - do not modify
private javax.swing.JLabel bobotLabel;
private javax.swing.JButton deleteButton;
private javax.swing.JTextField idKriteriaField;
private javax.swing.JLabel idKriteriaLabel;
private javax.swing.JLabel idVariabelLabel1;
private javax.swing.JLabel idVariabelLabel2;
private javax.swing.JFormattedTextField jFormattedTextField1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable masterTable;
private javax.swing.JButton newButton;
private javax.swing.JTextField parameterField;
private javax.swing.JLabel parameterLabel;
private javax.swing.JButton refreshButton;
private javax.swing.JButton saveButton;
private org.uin.spk.util.ZCombo zCombo1;
private org.uin.spk.util.ZCombo zCombo2;
// End of variables declaration
}

22. PemeriksaanPanel.java
/*
 * PemeriksaanPanel.java
 *
 * Created on Feb 15, 2011, 5:56:49 PM
 */
package org.uin.spk.screen;

import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import javax.persistence.EntityManager;
import javax.swing.JComboBox;
import javax.swing.JComponent;
import javax.swing.JComponent;
import javax.swing.JOptionPane;
import org.uin.spk.entity.KriteriaNilai;
import org.uin.spk.entity.MasterPerhitungan;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.Perhitungan;
import org.uin.spk.entity.PerhitunganPK;
import org.uin.spk.entity.controller.MasterPerhitunganJpaController;
import org.uin.spk.entity.controller.PerhitunganJpaController;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author haris_alkwitang
 */
public class PemeriksaanPanel extends javax.swing.JPanel {

EntityManager entityManager = Function.getEntityManager();
Pemohon pemohonSelected = null;
PerhitunganJpaController perhitunganController = new PerhitunganJpaController();
MasterPerhitunganJpaController masterPerhitunganController = new MasterPerhitunganJpaController();

/** Creates new form PemeriksaanPanel */
public PemeriksaanPanel() {
    initComponents();
    loadCombo();
    setLabel();
    setSelectedToNull(new JComponent[]{zCombo1, zCombo2, zCombo3, zCombo4, zCombo5, zCombo6, zCombo7, zCombo8, zCombo9, zCombo10, zCombo11, zCombo12, zCombo13, zCombo14, zCombo15, zCombo16, zCombo17, zCombo18});
}

private void setLabel() {
    jLabel1.setText(Function.getResources("label.nilaijaminan"));
    jLabel2.setText(Function.getResources("label.perilakusehari"));
    jLabel3.setText(Function.getResources("label.penampilan"));
    jLabel4.setText(Function.getResources("label.interaksimasyarakat"));
    jLabel5.setText(Function.getResources("label.ketaatanberibadah"));
    jLabel6.setText(Function.getResources("label.modal"));
    jLabel7.setText(Function.getResources("label.omzet"));
    jLabel8.setText(Function.getResources("label.lokasiusaha"));
    jLabel9.setText(Function.getResources("label.jenistempatusaha"));
    jLabel10.setText(Function.getResources("label.penghasilanbulan"));
    jLabel11.setText(Function.getResources("label.pengeluaranbulan"));
    jLabel12.setText(Function.getResources("label.persaingan"));
    jLabel13.setText(Function.getResources("label.lamausaha"));
    jLabel14.setText(Function.getResources("label.kepemilikantempatusaha"));
    jLabel15.setText(Function.getResources("label.jumlahtenagakerja"));
    jLabel16.setText(Function.getResources("label.kepemilikantempattinggal"));
}

```

```

jLabel17.setText(Function.getResources("label.kondisitempattinggal"));
jLabel18.setText(Function.getResources("label.calon"));
jLabel19.setText(Function.getResources("label.tanggungan"));

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(
Function.getResources("label.jaminan")));
jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(
Function.getResources("label.karakter")));
jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(
Function.getResources("label.usaha")));
jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(
Function.getResources("label.kemampuan")));

jButton1.setText(Function.getResources("button.pilihcalon"));
jButton2.setText(Function.getResources("button.periksa"));
jButton3.setText(Function.getResources("button.reset"));

private void loadCombo() {
zCombo1.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.nilajaminan"))));
zCombo2.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.perilaku"))));
zCombo3.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.penampilan"))));
zCombo4.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.interaksi"))));
zCombo5.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.ketaatan"))));
zCombo6.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.modalinvestasi"))));
zCombo7.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.omzet"))));
zCombo8.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.lokasiusaha"))));
zCombo9.setNilai(getComboValue(Integer.valueOf(Function.getR
esources("key.child.jenistempatusahaan"))));
zCombo10.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.penghasilan"))));
zCombo11.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.lamausaha"))));
zCombo12.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.persaingan"))));
zCombo13.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.kepemilikantempatusa"))));
zCombo14.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.jumlahtenagakerja"))));
zCombo15.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.pengeluaran"))));
zCombo16.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.kepemilikantempattinggal"))));
zCombo17.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.kondisitempattinggal"))));
zCombo18.setNilai(getComboValue(Integer.valueOf(Function.get
Resources("key.child.jumlahtanggungan"))));

setSelectedToNull(new JComponent[]{zCombo1, zCombo2,
zCombo3, zCombo4, zCombo5, zCombo6,
zCombo7, zCombo8, zCombo9, zCombo10,
zCombo11, zCombo12, zCombo13, zCombo14, zCombo15,
zCombo16,
zCombo17, zCombo18});
}

private void setSelectedToNull(JComponent listComponent[]) {
for (JComponent component : listComponent) {
if (component instanceof JComboBox) {
((JComboBox) component).setSelectedIndex(-1);
}
}
}

```

```

}

private HashMap getComboValue(Integer param) {
HashMap mapNilai = new HashMap();
List<KriteriaNilai> listNilai =
entityManager.createNamedQuery("KriteriaNilai.findByIdKriteriaInduk").setParameter("idKriteriaInduk", param).getResultSet();
for (KriteriaNilai kriteriaNilai : listNilai) {
mapNilai.put(kriteriaNilai.getParameter(), kriteriaNilai.getBobot());
}
return mapNilai;
}

private boolean doCheck(){
return checkSelected(new JComponent[]{zCombo1, zCombo2, zCombo3, zCombo4, zCombo5, zCombo6, zCombo7, zCombo8, zCombo9, zCombo10, zCombo11, zCombo12, zCombo13, zCombo14, zCombo15, zCombo16, zCombo17, zCombo18});
}

private boolean checkSelected(JComponent listComponent[]){
for(JComponent component : listComponent){
if(component instanceof JComboBox){
if(((JComboBox) component).getSelectedIndex() == -1){
Function.showError("alert.isi.periksa");
return false;
}
}
}
return true;
}

private void doPeriksa(){
Date currentDate = new Date();
MasterPerhitungan masterPerhitungan = new MasterPerhitungan();

masterPerhitungan.setNoCalonDebitur(pemohonSelected.getNoCalonDebitur());
masterPerhitungan.setDateCheck(currentDate);
masterPerhitunganController.create(masterPerhitungan);
masterPerhitungan = (MasterPerhitungan)
entityManager.createNamedQuery("MasterPerhitungan.findByNoCalonDebiturAndDateCheck").setParameter("dateCheck", currentDate).setParameter("noCalonDebitur", pemohonSelected.getNoCalonDebitur()).getSingleResult();
//start-Nilai Jaminan
Perhitungan phtNilaiJaminan = new Perhitungan(
new PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan(), Integer.valueOf(Function.getResources("key.child.nilajaminan"))));
phtNilaiJaminan.setNilai(Double.parseDouble(zCombo1.getSelectedKey().toString()));
//end-Nilai Jaminan
//start-Nilai Perilaku
Perhitungan phtPerilaku = new Perhitungan(
new PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan(), Integer.valueOf(Function.getResources("key.child.perilaku"))));
phtPerilaku.setNilai(Double.parseDouble(zCombo2.getSelectedKey().toString()));
//end-Nilai Perilaku
//start-Penampilan
Perhitungan phtPenampilan = new Perhitungan(
new PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan(), Integer.valueOf(Function.getResources("key.child.penampilan"))));

```

```

Integer.valueOf(Function.getResources("key.child.penampilan")));
;

phtPenampilan.setNilai(Double.parseDouble(zCombo3.getSelectedKey().toString()));
//end-Penampilan
//start-Interaksi
Perhitungan phtInteraksi = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.interaksi")));
    phtInteraksi.setNilai(Double.parseDouble(zCombo4.getSelectedKey().toString()));
//end-Interaksi
//start-Ketaatan
Perhitungan phtKetaatan = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.ketaatan")));
    phtKetaatan.setNilai(Double.parseDouble(zCombo5.getSelectedKey().toString()));
//end-Ketaatan
//start-Modal Investasi
Perhitungan phtModalInves = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.modalinvestasi")));
    phtModalInves.setNilai(Double.parseDouble(zCombo6.getSelectedKey().toString()));
//end-Modal Investasi
//start-Omzet
Perhitungan phtOmzet = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.omzet")));
    phtOmzet.setNilai(Double.parseDouble(zCombo7.getSelectedKey().toString()));
//end-Omzet
//start-Lokasi Usaha
Perhitungan phtLokasiUsaha = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.lokasiusaha")));
;

phtLokasiUsaha.setNilai(Double.parseDouble(zCombo8.getSelectedKey().toString()));
//end-Lokasi Usaha
//start-Jenis Tempat Usaha
Perhitungan phtJenisTempatUsaha = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.jenistempatusaha")));
;

phtJenisTempatUsaha.setNilai(Double.parseDouble(zCombo9.getSelectedKey().toString()));
//end-Jenis Tempat Usaha
//start-Penghasilan
Perhitungan phtPenghasilan = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.penghasilan")));
;

phtPenghasilan.setNilai(Double.parseDouble(zCombo10.getSelectedKey().toString()));
//end-Penghasilan
//start-Lama Usaha
Perhitungan phtLamaUsaha = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),

Integer.valueOf(Function.getResources("key.child.lamausaha")));
;

phtLamaUsaha.setNilai(Double.parseDouble(zCombo11.getSelectedKey().toString()));
//end-Lama Usaha
//start-Persaingan
Perhitungan phtPersaingan = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.persaingan")));
;

phtPersaingan.setNilai(Double.parseDouble(zCombo12.getSelectedKey().toString()));
//end-Persaingan
//start-Kepemilikan Tempat Usaha
Perhitungan phtKepemilikanTmpatUsaha = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.kepemilikantempatusaha")));
;

phtKepemilikanTmpatUsaha.setNilai(Double.parseDouble(zCombo13.getSelectedKey().toString()));
//end-Kepemilikan Tempat Usaha
//start-Jumlah Tenaga Kerja
Perhitungan phtJumlahTenagaKerja = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.jumlahtenagakerja")));
;

phtJumlahTenagaKerja.setNilai(Double.parseDouble(zCombo14.getSelectedKey().toString()));
//end-Jumlah Tenaga Kerja
//start-Pengeluaran
Perhitungan phtPengeluaran = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.pengeluaran")));
;

phtPengeluaran.setNilai(Double.parseDouble(zCombo15.getSelectedKey().toString()));
//end-Pengeluaran
//start-Kepemilikan Tempat Tinggal
Perhitungan phtKpmilanTmpatTinggal = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.kepamilikantinggal")));
;

phtKpmilanTmpatTinggal.setNilai(Double.parseDouble(zCombo16.getSelectedKey().toString()));
//end-Kepemilikan Tempat Tinggal
//start-Kondisi Tempat Tinggal
Perhitungan phtKondisiTmpatTinggal = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.kondisitempattinggal")));
;

phtKondisiTmpatTinggal.setNilai(Double.parseDouble(zCombo17.getSelectedKey().toString()));
//end-Kondisi Tempat Tinggal
//start-Jumlah Tanggungan
Perhitungan phtJumlahTanggungan = new Perhitungan(
    new
    PerhitunganPK(masterPerhitungan.getIdMasterPerhitungan()),
    Integer.valueOf(Function.getResources("key.child.jumlahtanggungan")));
;

phtJumlahTanggungan.setNilai(Double.parseDouble(zCombo18.getSelectedKey().toString()));

```

```

//end-Jumlah Tanggungan

List<Perhitungan> listPerhitungan = new
ArrayList<Perhitungan>();
listPerhitungan.add(phtNilaiJaminan);
listPerhitungan.add(phtPerilaku);
listPerhitungan.add(phtPenampilan);
listPerhitungan.add(phtInteraksi);
listPerhitungan.add(phtKetaatan);
listPerhitungan.add(phtModalInves);
listPerhitungan.add(phtOmzet);
listPerhitungan.add(phtLokasiUsaha);
listPerhitungan.add(phtJenisTempatUsaha);
listPerhitungan.add(phtPenghasilan);
listPerhitungan.add(phtLamaUsaha);
listPerhitungan.add(phtPersaingan);
listPerhitungan.add(phtKepemilikanTmpatUsaha);
listPerhitungan.add(phtJumlahTenagaKerja);
listPerhitungan.add(phtPengeluaran);
listPerhitungan.add(phtKpmilanTmpatTinggal);
listPerhitungan.add(phtKondisiTmpatTinggal);
listPerhitungan.add(phtJumlahTanggungan);
try {
    perhitunganController.create(listPerhitungan);
} catch (Exception e) {
    e.printStackTrace();
}
MainFrame.setPanel(new
PerhitunganPanel(masterPerhitungan));
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    zCombo1 = new org.uin.spk.util.ZCombo();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    zCombo2 = new org.uin.spk.util.ZCombo();
    jLabel3 = new javax.swing.JLabel();
    zCombo3 = new org.uin.spk.util.ZCombo();
    zCombo4 = new org.uin.spk.util.ZCombo();
    jLabel4 = new javax.swing.JLabel();
    zCombo5 = new org.uin.spk.util.ZCombo();
    jLabel5 = new javax.swing.JLabel();
    jPanel3 = new javax.swing.JPanel();
    jLabel6 = new javax.swing.JLabel();
    zCombo6 = new org.uin.spk.util.ZCombo();
    jLabel7 = new javax.swing.JLabel();
    zCombo7 = new org.uin.spk.util.ZCombo();
    zCombo8 = new org.uin.spk.util.ZCombo();
    jLabel8 = new javax.swing.JLabel();
    zCombo9 = new org.uin.spk.util.ZCombo();
    jLabel9 = new javax.swing.JLabel();
    zCombo11 = new org.uin.spk.util.ZCombo();
    jLabel12 = new javax.swing.JLabel();
    zCombo12 = new org.uin.spk.util.ZCombo();
    jLabel13 = new javax.swing.JLabel();
    zCombo13 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    zCombo13 = new org.uin.spk.util.ZCombo();
    zCombo14 = new org.uin.spk.util.ZCombo();
    jLabel15 = new javax.swing.JLabel();
    jPanel4 = new javax.swing.JPanel();
    jLabel10 = new javax.swing.JLabel();
    zCombo10 = new org.uin.spk.util.ZCombo();
    jLabel11 = new javax.swing.JLabel();
    zCombo11 = new org.uin.spk.util.ZCombo();
    zCombo15 = new org.uin.spk.util.ZCombo();
}

zCombo16 = new org.uin.spk.util.ZCombo();
jLabel16 = new javax.swing.JLabel();
zCombo17 = new org.uin.spk.util.ZCombo();
jLabel17 = new javax.swing.JLabel();
zCombo18 = new org.uin.spk.util.ZCombo();
jLabel19 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jTextField2 = new javax.swing.JTextField();
jButton3 = new javax.swing.JButton();

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(""
Jaminan"));

jLabel1.setText("Nilai Jaminan");

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel1)
.addComponent(zCombo1)
)
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
PREFERRED_SIZE, 175,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(319, Short.MAX_VALUE)
);

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
LEADING)
);

.jaddGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
BASELINE)
.addComponent(jLabel11)
.addComponent(zCombo1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(""
Karakter));

jLabel2.setText("Perilaku sehari-hari");
jLabel3.setText("Penampilan");
jLabel4.setText("Interaksi Masyarakat");
jLabel5.setText("Ketaatan Beribadah");

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addComponent(jLabel2)
.addComponent(zCombo2)
)
);
jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
PREFERRED_SIZE, 175,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(319, Short.MAX_VALUE)
);

jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
TRAILING)
.addComponent(jLabel2)
.addComponent(jLabel3)
);

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)

```





```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(zCombo10,
javax.swing.GroupLayout.PREFERRED_SIZE, 144,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addComponent(jLabel19)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(zCombo18,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel4Layout.createSequentialGroup()
    .addComponent(jLabel17)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(zCombo17,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(zCombo15,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addComponent(jLabel16)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(zCombo16,
javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(13, Short.MAX_VALUE))
);

jPanel4Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL,
AL, new java.awt.Component[] {jLabel10, jLabel11, jLabel16,
jLabel17, jLabel19});

jPanel4Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL,
AL, new java.awt.Component[] {zCombo10, zCombo15,
zCombo16});

jPanel4Layout.setVerticalGroup(
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel10)
    .addComponent(zCombo10,
javax.swing.GroupLayout.PREFERRED_SIZE,
```



```

        doPeriksa();
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private org.uin.spk.util.ZCombo zCombo1;
    private org.uin.spk.util.ZCombo zCombo10;
    private org.uin.spk.util.ZCombo zCombo11;
    private org.uin.spk.util.ZCombo zCombo12;
    private org.uin.spk.util.ZCombo zCombo13;
    private org.uin.spk.util.ZCombo zCombo14;
    private org.uin.spk.util.ZCombo zCombo15;
    private org.uin.spk.util.ZCombo zCombo16;
    private org.uin.spk.util.ZCombo zCombo17;
    private org.uin.spk.util.ZCombo zCombo18;
    private org.uin.spk.util.ZCombo zCombo2;
    private org.uin.spk.util.ZCombo zCombo3;
    private org.uin.spk.util.ZCombo zCombo4;
    private org.uin.spk.util.ZCombo zCombo5;
    private org.uin.spk.util.ZCombo zCombo6;
    private org.uin.spk.util.ZCombo zCombo7;
    private org.uin.spk.util.ZCombo zCombo8;
    private org.uin.spk.util.ZCombo zCombo9;
    // End of variables declaration
}

```

### 23. PemohonDialog.java

```

/*
 * PemohonDialog.java
 *
 * * Created on Feb 15, 2011, 6:54:50 PM
 */

package org.uin.spk.screen;

import java.util.List;
import javax.swing.JComponent;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.controller.PemohonJpaController;
import org.uin.spk.util.Function;

/**
 *
 * @author haris_alkwitang
 */
public class PemohonDialog extends javax.swing.JDialog {

```

```

PemohonJpaController pemohonController = new
PemohonJpaController();
Pemohon pemohonSelected;

/** Creates new form PemohonDialog */
public PemohonDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    refresh();
}

private void setHeaderTable() {
    DefaultTableModel tableModel = new DefaultTableModel();

    tableModel.addColumn(Function.getResources("table.header.nope
mohon"));

    tableModel.addColumn(Function.getResources("table.header.jenis
pemohon"));

    tableModel.addColumn(Function.getResources("table.header.nama
pemohon"));

    tableModel.addColumn(Function.getResources("table.header.alam
atpemohon"));
    jTable1.setModel(tableModel);
}

private void refresh() {
    setHeaderTable();
    DefaultTableModel dtm = (DefaultTableModel)
jTable1.getModel();
    List<Pemohon> listPemohon =
pemohonController.findPemohonEntities();
    for (Pemohon pemohon : listPemohon) {
        dtm.addRow(new Object[]{pemohon.getNoCalonDebitur(),
pemohon.getJenisPemohon(), pemohon.getNamaCalonDebitur(),
pemohon.getAlamat()});
    }
}

public Pemohon getPemohonSelected(){
    return pemohonSelected;
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new JTable(){
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOS
E_ON_CLOSE);

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {},
            {}
        },
        new String [] {
            ""
        }
    ));
    jTable1.setToolTipText("Double Click to Choose Pemohon");
    jTable1.addMouseListener(new
java.awt.event.MouseAdapter() {

```

```

        public void mouseClicked(java.awt.event.MouseEvent evt)
    {
        jTable1MouseClicked(evt);
    }
};

jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 471,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 399,
Short.MAX_VALUE)
    );
}

pack();
}// </editor-fold>

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if(evt.getClickCount() == 2){
        pemohonSelected =
pemohonController.findPemohon(jTable1.getValueAt(jTable1.getSelectedRow(), 0).toString());
        this.dispose();
    }
}

// Variables declaration - do not modify
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

**24. PemohonPanel.java**

```

/*
 * PemohonPanel.java
 *
 * Created on Feb 15, 2011, 4:31:10 PM
 */
package org.uin.spk.screen;

import java.io.File;
import java.util.List;
import javax.swing.JComponent;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.controller.PemohonJpaController;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author Haris_alkwitang
 */
public class PemohonPanel extends javax.swing.JPanel {

    PemohonJpaController pemohonController = new
    PemohonJpaController();

```

```

FileNameExtensionFilter filter = new
FileNameExtensionFilter("JPEG file", "jpg", "jpeg");
JFileChooser pilihGambar = new JFileChooser();
File file = null;
byte[] byteImage = null;

/** Creates new form PemohonPanel */
public PemohonPanel() {
    initComponents();
    setLabel();
    refresh();
}

private void setLabel() {
    saveButton.setText(Function.getResources("button.save"));
    newButton.setText(Function.getResources("button.new"));

    deleteButton.setText(Function.getResources("button.delete"));
    refreshButton.setText(Function.getResources("button.refresh"));
    jLabel1.setText(Function.getResources("label.nopemohon"));

    jLabel2.setText(Function.getResources("label.jenispemohon"));
    jLabel3.setText(Function.getResources("label.nama"));
    jLabel4.setText(Function.getResources("label.alamatrumah"));
    jLabel5.setText(Function.getResources("label.pekerjaan"));
    jLabel6.setText(Function.getResources("label.rencanapembimbingan"));
    jLabel7.setText(Function.getResources("label.nilaijinjaman"));
    jLabel8.setText(Function.getResources("label.jangkawaktupembimbingan"));
    jLabel9.setText(Function.getResources("label.jenisbarangjaminan"));
    jLabel10.setText(Function.getResources("label.nilaitaksasijaminan"));
    jLabel11.setText(Function.getResources("label.penyerahanadministrasi"));
    jLabel12.setText(Function.getResources("label.rp"));
    jLabel13.setText(Function.getResources("label.rp"));
    jLabel14.setText(Function.getResources("label.fotopemohon"));
    jRadioButton1.setText(Function.getResources("label.lama"));
    jRadioButton2.setText(Function.getResources("label.baru"));
    jRadioButton3.setText(Function.getResources("label.memenuhi"));
    jRadioButton4.setText(Function.getResources("label.tidakmemenuhi"));
    jButton1.setText(Function.getResources("button.pilihfile"));
    jButton2.setText(Function.getResources("button.preview"));
}

private void setHeaderTable() {
    DefaultTableModel tableModel = new DefaultTableModel();

    tableModel.addColumn(Function.getResources("table.header.nopemohon"));
    tableModel.addColumn(Function.getResources("table.header.jenispemohon"));
    tableModel.addColumn(Function.getResources("table.header.nama"));
    jTable1.setModel(tableModel);
}

private void refresh() {
    setHeaderTable();
    DefaultTableModel dtm = (DefaultTableModel)
jTable1.getModel();
    List<Pemohon> listPemohon =
pemohonController.findPemohonEntities();
    for (Pemohon pemohon : listPemohon) {
        dtm.addRow(new Object[]{pemohon.getNoCalonDebitur(),
pemohon.getJenisPemohon(), pemohon.getNamaCalonDebitur()});
    }
    Function.clearComponent(new JComponent[]{jTextField1,
jTextField2, jTextField3, jTextField4, jTextField5, jTextField6,
jFormattedTextField1, jFormattedTextField2, jComboBox1,
jComboBox2, jComboBox3});
    jRadioButton1.setSelected(true);
    jRadioButton3.setSelected(true);
}

```



```

buttonGroup2.add(jRadioButton4);
jRadioButton4.setText("Tidak Memenuhi");

jLabel14.setText("Foto Pemohon");
jTextField6.setEditable(false);

jButton1.setText("Choose File");
jButton1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Preview");
jButton2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton2ActionPerformed(evt);
    }
});

newButton.setText("New");
newButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        newButtonActionPerformed(evt);
    }
});

deleteButton.setText("Delete");
deleteButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        deleteButtonActionPerformed(evt);
    }
});

refreshButton.setText("Refresh");
refreshButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        refreshButtonActionPerformed(evt);
    }
});

saveButton.setText("Save");
saveButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        saveButtonActionPerformed(evt);
    }
});

jLabel15.setText("Akad");

jComboBox3.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Al-
Murabahah", "Al-Mudharabah", "Musyarakah", " " }));
jComboBox3.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jComboBox3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1):

```

```

        .addComponent(jTextField3,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE)
        .addComponent(jTextField4,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE)
        .addComponent(jTextField2,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE)
        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE, 211,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jFormattedTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 195,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jFormattedTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 195,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE, 211,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField5,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE)
        .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 183,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jRadioButton3,
javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jTextField3,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE)
        .addComponent(jRadioButton4,
javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, 166,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jButton1)

.addComponent(jButton2)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jRadioButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 56,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jRadioButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 427,
Short.MAX_VALUE))
        .addComponent(containerGap())

.addGroup(jPanel1Layout.createSequentialGroup()
    .addContainerGap()
    .addComponent(newButton))

.addComponent(deleteButton)

.addComponent(refreshButton)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(saveButton))
);

jPanel1Layout.linkSize(javax.swing.SwingConstantsConstants.HORIZONTAL,
AL, new java.awt.Component[] {jLabel14, jLabel3});

jPanel1Layout.linkSize(javax.swing.SwingConstantsConstants.HORIZONTAL,
AL, new java.awt.Component[] {jComboBox1, jComboBox3});

jPanel1Layout.linkSize(javax.swing.SwingConstantsConstants.HORIZONTAL,
AL, new java.awt.Component[] {jLabel15, jLabel5, jLabel6});

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 14,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jRadioButton1)
                .addComponent(jRadioButton2))

        .addComponent(jTextField3,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton1)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE),
            .addComponent(jButton2))

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton1)
            .addComponent(jButton2))

        .addComponent(jTextField3,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton2)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel5)
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE),
            .addComponent(jButton1)
            .addComponent(jButton2))

        .addComponent(jTextField3,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton2)

```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel15)
        .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel6)
        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel7)
        .addComponent(jFormattedTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel12))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel8)
        .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel9)
        .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel10)
        .addComponent(jFormattedTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
pemohon.setAkad(jComboBox3.getSelectedItem().toString());
    pemohonController.create(pemohon);
} else {
    pemohon =
pemohonController.findPemohon(jTextField1.getText());

pemohon.setJenisPemohon(jRadioButton1.isSelected() ?
Function.getResources("label.lama") :
Function.getResources("label.baru"));

pemohon.setNamaCalonDebitur(jTextField2.getText());
    pemohon.setAlamat(jTextField3.getText());
    pemohon.setPekerjaan(jTextField4.getText());

pemohon.setRencanaPenggunaan(jComboBox1.getSelectedItem().t
oString());

pemohon.setNilaiJaminan(String.valueOf(jFormattedTextField1.ge
tValue()));

pemohon.setJangkaWaktuBulan(jTextField5.getText());

pemohon.setJenisJaminan(jComboBox2.getSelectedItem().toString()
);

pemohon.setNilaiTaksasi(String.valueOf(jFormattedTextField2.get
Value()));
    pemohon.setAdministrasi(jRadioButton3.isSelected()
? Function.getResources("label.memenuhi") :
Function.getResources("label.tidakmemenuhi"));
    if (file != null) {

pemohon.setPemohonimage(Function.getBytesFromFile(file));
    }

pemohon.setAkad(jComboBox3.getSelectedItem().toString());
    pemohonController.edit(pemohon);
}
refresh();
} catch (Exception e) {
    e.printStackTrace();
}
}

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    pilihGambar.addChoosableFileFilter(filter);
    if (JFileChooser.APPROVE_OPTION ==
pilihGambar.showOpenDialog(null)) {
        if (pilihGambar.isAcceptAllFileFilterUsed()) {
            file = pilihGambar.getSelectedFile();
            try {
                byteImage = Function.getBytesFromFile(file);
            } catch (Exception e) {
                e.printStackTrace();
            }
            jTextField6.setText(file.getPath());
        }
    }
}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ImageDialog image = null;
    if (file != null) {
        try {
            image = new ImageDialog(GlobVar.mainFrame, true,
Function.getBytesFromFile(file));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    if (byteImage != null) {
        try {
```

```

        image = new ImageDialog(GlobVar.mainFrame, true,
byteImage);
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    image = new ImageDialog(GlobVar.mainFrame, true);
}
image.setLocationRelativeTo(null);
image.setVisible(true);
image.repaint();
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    binding();
}

private void jTable1KeyReleased(java.awt.event.KeyEvent evt) {
    binding();
}

private void jComboBox3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup2;
private javax.swing.JButton deleteButton;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JComboBox jComboBox3;
private javax.swing.JFormattedTextField jFormattedTextField1;
private javax.swing.JFormattedTextField jFormattedTextField2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JPanel jPanel1;
private javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton2;
private javax.swing.JRadioButton jRadioButton3;
private javax.swing.JRadioButton jRadioButton4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JButton newButton;
private javax.swing.JButton refreshButton;
private javax.swing.JButton saveButton;
// End of variables declaration
}

/*
 * Created on Feb 15, 2011, 5:56:49 PM
 */
package org.uin.spk.screen;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import javax.persistence.EntityManager;
import org.uin.spk.entity.MasterPerhitungan;
import org.uin.spk.entity.Pemohon;
import org.uin.spk.entity.Perhitungan;
import org.uin.spk.entity.controller.MasterPerhitunganJpaController;
import org.uin.spk.entity.controller.PemohonJpaController;
import org.uin.spk.entity.controller.PerhitunganJpaController;
import org.uin.spk.logic.FuzzyLogic;
import org.uin.spk.logic.Kriteria;
import org.uin.spk.util.Function;
import org.uin.spk.util.GlobVar;

/**
 *
 * @author haris_alkwitang
 */
public class PerhitunganPanel extends javax.swing.JPanel {

    EntityManager entityManager = Function.getEntityManager();
    MasterPerhitungan masterPerhitunganSelected;
    Pemohon pemohonSelected;
    PerhitunganJpaController perhitunganController = new PerhitunganJpaController();
    MasterPerhitunganJpaController masterPerhitunganController = new MasterPerhitunganJpaController();
    PemohonJpaController pemohonController = new PemohonJpaController();

    /** Creates new form PemeriksaanPanel */
    public PerhitunganPanel() {
        initComponents();
        setLabel();
    }

    public PerhitunganPanel(MasterPerhitungan masterPerhitungan) {
        initComponents();
        setLabel();
        doFill(masterPerhitungan);
    }

    private void setLabel() {
        jLabel1.setText(Function.getResources("label.nilaijaminan"));
        jLabel2.setText(Function.getResources("label.perilakusehari"));
        jLabel3.setText(Function.getResources("label.penampilan"));
        jLabel4.setText(Function.getResources("label.interaksimasyarakat"));
        jLabel5.setText(Function.getResources("label.ketaatanberibadah"));
        jLabel6.setText(Function.getResources("label.modal"));
        jLabel7.setText(Function.getResources("label.omzet"));
        jLabel8.setText(Function.getResources("label.lokasiusaha"));
        jLabel9.setText(Function.getResources("label.jenistempatusaha"));
        jLabel10.setText(Function.getResources("label.penghasilanbulan"));
        jLabel11.setText(Function.getResources("label.pengeluaranbulan"));
        jLabel12.setText(Function.getResources("label.persaingan"));
        jLabel13.setText(Function.getResources("label.lamausaha"));
        jLabel14.setText(Function.getResources("label.kepemilikantempatusaha"));
        jLabel15.setText(Function.getResources("label.jumlahtenagakerja"));
        jLabel16.setText(Function.getResources("label.kepemilikantempattinggal"));
        jLabel17.setText(Function.getResources("label.kondisitempattinggg"));
    }
}

```

**25. PerhitunganPanel.java**

```

/*
 * PemeriksaanPanel.java
 */

```

```

al"));
    jLabel18.setText(Function.getResources("label.calon"));
    jLabel19.setText(Function.getResources("label.tanggungan"));
    jLabel20.setText(Function.getResources("label.jumlahjaminan"));
    jLabel21.setText(Function.getResources("label.jumlahhusaha"));
    jLabel22.setText(Function.getResources("label.jumlahkemampuan"));
    jLabel23.setText(Function.getResources("label.jumlahkarakter"));
    jLabel24.setText(Function.getResources("label.date"));
    jLabel25.setText(Function.getResources("label.hasilpenilaian"));
    jLabel26.setText(Function.getResources("label.rekomendasi"));
    jLabel27.setText(Function.getResources("label.jumlahpinjamandib
erikan"));
}

private void doChoose() {
    MasterPerhitunganDialog masterDialog = new
MasterPerhitunganDialog(GlobVar.mainFrame, true);
    masterDialog.setLocationRelativeTo(null);
    masterDialog.setVisible(true);
    doFill(masterDialog.getMasterPerhitunganSelected());
}

private void doFill(MasterPerhitungan masterPerhitungan) {
    masterPerhitunganSelected = masterPerhitungan;
    pemohonSelected =
pemohonController.findPemohon(masterPerhitunganSelected.getN
oCalonDebitur());
    jTextField1.setText(pemohonSelected.getNoCalonDebitur() + +
");
    jTextField2.setText(pemohonSelected.getNamaCalonDebitur());
    jFormattedTextField23.setValue(masterPerhitunganSelected.getDa
taCheck());
    jFormattedTextField25.setValue(Double.valueOf(pemohonSelecte
d.getNilaiJaminan()));

    List<Perhitungan> listPerhitungan =
entityManager.createNamedQuery("Perhitungan.findByIdMasterP
erhitungan").setParameter("idMasterPerhitungan",
masterPerhitunganSelected.getIdMasterPerhitungan()).getResul
tList();
    for (Perhitungan perhitungan : listPerhitungan) {
        if (perhitungan.getPerhitunganPK().getIdKriteria() ==
Integer.valueOf(Function.getResources("key.child.nilaijaminan")))
    }

    jFormattedTextField1.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
    } else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.penghasilan")))
    {

    jFormattedTextField14.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
    } else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.pengeluaran")))
    {

    jFormattedTextField15.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
    } else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.kepemilikantem
pattinggal")))
    {

    jFormattedTextField16.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
    } else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.kondisitempati
nggal")))
    }

jFormattedTextField17.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.jumlahtanggung
an")))
{

jFormattedTextField18.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.modalinvestasi"))
)
{

jFormattedTextField6.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.omzet")))
{

jFormattedTextField7.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.lokasiusaha")))
{

jFormattedTextField8.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.jenistempatusah
a")))
{

jFormattedTextField9.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.lamausaha")))
{

jFormattedTextField10.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.persaingan")))
{

jFormattedTextField11.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.kepemilikantem
patusaha")))
{

jFormattedTextField12.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.jumlahtenagake
rja")))
{

jFormattedTextField13.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.perilaku")))
{

jFormattedTextField2.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
} else if (perhitungan.getPerhitunganPK().getIdKriteria()
===
Integer.valueOf(Function.getResources("key.child.penampilan")))
{

jFormattedTextField3.setValue(Double.parseDouble(perhitungan.g
etNilai() + ""));
}

```

```

        } else if (perhitungan.getPerhitunganPK().getIdKriteria() == Integer.valueOf(Function.getResources("key.child.interaksi")))
    {
        jFormattedTextField4.setValue(Double.parseDouble(perhitungan.getNilai() + ""));
        } else if (perhitungan.getPerhitunganPK().getIdKriteria() == Integer.valueOf(Function.getResources("key.child.ketaatan")))
    {
        jFormattedTextField5.setValue(Double.parseDouble(perhitungan.getNilai() + ""));
    }
    doSumTextField();
    calculate();
}

private void doSumTextField()
{
jFormattedTextField19.setValue(jFormattedTextField1.getValue());
;

jFormattedTextField21.setValue(Double.parseDouble(jFormattedTextField14.getValue() +
+ Double.parseDouble(jFormattedTextField15.getValue() + "") +
Double.parseDouble(jFormattedTextField16.getValue() + "") +
+ Double.parseDouble(jFormattedTextField17.getValue() + "") +
Double.parseDouble(jFormattedTextField18.getValue() + ""));

jFormattedTextField22.setValue(Double.parseDouble(jFormattedTextField2.getValue() + "") +
+ Double.parseDouble(jFormattedTextField3.getValue() + "") +
Double.parseDouble(jFormattedTextField4.getValue() + "") +
+ Double.parseDouble(jFormattedTextField5.getValue() + ""));
;

jFormattedTextField20.setValue(Double.parseDouble(jFormattedTextField6.getValue() + "") +
+ Double.parseDouble(jFormattedTextField7.getValue() + "") +
Double.parseDouble(jFormattedTextField8.getValue() + "") +
+ Double.parseDouble(jFormattedTextField9.getValue() + "") +
Double.parseDouble(jFormattedTextField10.getValue() + "") +
+ Double.parseDouble(jFormattedTextField11.getValue() + "") +
Double.parseDouble(jFormattedTextField12.getValue() + "") +
+ Double.parseDouble(jFormattedTextField13.getValue() + ""));
;

private void calculate()
{
Kriteria kriteria = new Kriteria();

kriteria.setJaminan(Double.parseDouble(jFormattedTextField19.getValue() + ""));
kriteria.setKemampuan(Double.parseDouble(jFormattedTextField21.getValue() + ""));
kriteria.setUsaha(Double.parseDouble(jFormattedTextField20.getValue() + ""));
kriteria.setKarakter(Double.parseDouble(jFormattedTextField22.getValue() + ""));
FuzzyLogic fuzzyLogic = new FuzzyLogic();
double fuzzyValue = fuzzyLogic.calculate(kriteria);
jFormattedTextField24.setValue(fuzzyValue);
FuzzyLogic.OutputRange outputRange =
fuzzyLogic.checkRangeVariableOutput(fuzzyValue);
}
}

if
(outputRange.equals(FuzzyLogic.OutputRange.SANGAT_LAYAK))
{
    jTextField3.setText(Function.getResources("system.sl"));
} else if
(outputRange.equals(FuzzyLogic.OutputRange.LAYAK_DENGA_N_RESIKO_RENDAH))
{
    jTextField3.setText(Function.getResources("system.lrr"));
} else if
(outputRange.equals(FuzzyLogic.OutputRange.LAYAK_DENGA_N_RESIKO_TINGGI))
{
    jTextField3.setText(Function.getResources("system.lrt"));
} else if
(outputRange.equals(FuzzyLogic.OutputRange.KURANG_LAYAK))
{
    jTextField3.setText(Function.getResources("system.kl"));
} else if
(outputRange.equals(FuzzyLogic.OutputRange.SANGAT_LAYAK))
{
    jTextField3.setText(Function.getResources("system.stl"));
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents()
{
jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jFormattedTextField1 = new javax.swing.JFormattedTextField();
jLabel20 = new javax.swing.JLabel();
jFormattedTextField19 = new javax.swing.JFormattedTextField();
jPanel2 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jFormattedTextField2 = new javax.swing.JFormattedTextField();
jFormattedTextField3 = new javax.swing.JFormattedTextField();
jFormattedTextField4 = new javax.swing.JFormattedTextField();
jFormattedTextField5 = new javax.swing.JFormattedTextField();
jLabel23 = new javax.swing.JLabel();
jFormattedTextField22 = new javax.swing.JFormattedTextField();
jPanel3 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jFormattedTextField6 = new javax.swing.JFormattedTextField();
jFormattedTextField7 = new javax.swing.JFormattedTextField();
jFormattedTextField8 = new javax.swing.JFormattedTextField();
jFormattedTextField9 = new javax.swing.JFormattedTextField();
jFormattedTextField10 = new javax.swing.JFormattedTextField();
}

```



```
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00")));

jLabel23.setFont(new java.awt.Font("Tahoma", 1, 11));
jLabel23.setText("Jumlah Karakter");

jFormattedTextField22.setEditable(false);
jFormattedTextField22.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));
jFormattedTextField22.setFont(new java.awt.Font("Tahoma",
1, 11));

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel2Layout.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE).addGroup(jPanel2Layout.createSequentialGroup().addComponent(jLabel3).addComponent(jLabel4).addComponent(jLabel2).addComponent(jLabel23).addComponent(jLabel5)).addGroup(jPanel2Layout.createSequentialGroup().addComponent(jFormattedTextField22, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE)).addGap(50, Short.MAX_VALUE))).addContainerGap());
jPanel2Layout.setVerticalGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.createSequentialGroup().addGroup(jPanel2Layout.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE).addComponent(jLabel23).addComponent(jFormattedTextField22, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)).addGap(50, Short.MAX_VALUE)).addContainerGap()));

jPanel2Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jLabel2, jLabel23, jLabel3, jLabel4, jLabel5});

jPanel2Layout.setVerticalGroup();

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jPanel2Layout.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE).addComponent(jLabel2).addComponent(jLabel3).addComponent(jLabel4).addComponent(jLabel5)).addGroup(jPanel2Layout.createSequentialGroup().addComponent(jFormattedTextField6, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField7, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField8, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField9, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)).addGap(50, Short.MAX_VALUE))).addContainerGap());
jPanel2Layout.setVerticalGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE).addComponent(jLabel2).addComponent(jLabel3).addComponent(jLabel4).addComponent(jLabel5)).addGroup(jPanel2Layout.createSequentialGroup().addComponent(jFormattedTextField6, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField7, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField8, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jFormattedTextField9, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)).addGap(50, Short.MAX_VALUE)).addContainerGap());

jLabel6.setText("Modal Investasi");
jLabel7.setText("Omzet /bln");
jLabel8.setText("Lokasi Usaha");
jLabel9.setText("Jenis Tempat Usaha");
jLabel11.setText("Persaingan");
jLabel13.setText("Lama Usaha");
jLabel14.setText("Kepemilikan Tempat Usaha");
jLabel15.setText("Jumlah Tenaga Kerja");

jFormattedTextField6.setEditable(false);
jFormattedTextField6.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField7.setEditable(false);
jFormattedTextField7.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField8.setEditable(false);
jFormattedTextField8.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));
```



```
.addComponent(jLabel9)
.addComponent(jFormattedTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel13)
.addComponent(jFormattedTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel12)
.addComponent(jFormattedTextField11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel14)
.addComponent(jFormattedTextField12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel15)
.addComponent(jFormattedTextField13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel21)
.addComponent(jFormattedTextField20,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Kemampuan"));

jLabel10.setText("Penghasilan /bulan");
jLabel11.setText("Pengeluaran /bulan");
jLabel16.setText("Kepemilikan Tempat Tinggal");
jLabel17.setText("Kondisi Tempat Tinggal");
jLabel19.setText("Jumlah Tanggungan Keluarga");

jFormattedTextField14.setEditable(false);
```

```
jFormattedTextField14.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField15.setEditable(false);
jFormattedTextField15.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField16.setEditable(false);
jFormattedTextField16.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField17.setEditable(false);
jFormattedTextField17.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jFormattedTextField18.setEditable(false);
jFormattedTextField18.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));

jLabel22.setFont(new java.awt.Font("Tahoma", 1, 11));
jLabel22.setText("Jumlah Kemampuan");

jFormattedTextField21.setEditable(false);
jFormattedTextField21.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new
java.text.DecimalFormat("#0.00"))));
jFormattedTextField21.setFont(new java.awt.Font("Tahoma",
1, 11));

javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addComponent(jLabel19)
.addComponent(jLabel17)
.addComponent(jLabel16)
.addComponent(jLabel10)
.addComponent(jLabel11)
.addComponent(jLabel22))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jFormattedTextField15,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jFormattedTextField14,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jFormattedTextField18,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jFormattedTextField21,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addComponent(jFormattedTextField17,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jFormattedTextField16,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

jPanel4Layout.linkSize(javax.swing.SwingConstantsConstants.HORIZONTAL,
AL, new java.awt.Component[] {jLabel10, jLabel11, jLabel16,
jLabel17, jLabel19, jLabel22});

jPanel4Layout.setVerticalGroup(
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel10)
.addComponent(jFormattedTextField14,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel11)
.addComponent(jFormattedTextField15,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel16)
.addComponent(jFormattedTextField16,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel17)
.addComponent(jFormattedTextField17,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel19)
.addComponent(jFormattedTextField18,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel22));

```





```

private javax.swing.JFormattedTextField jFormattedTextField16;
private javax.swing.JFormattedTextField jFormattedTextField17;
private javax.swing.JFormattedTextField jFormattedTextField18;
private javax.swing.JFormattedTextField jFormattedTextField19;
private javax.swing.JFormattedTextField jFormattedTextField2;
private javax.swing.JFormattedTextField jFormattedTextField20;
private javax.swing.JFormattedTextField jFormattedTextField21;
private javax.swing.JFormattedTextField jFormattedTextField22;
private javax.swing.JFormattedTextField jFormattedTextField23;
private javax.swing.JFormattedTextField jFormattedTextField24;
private javax.swing.JFormattedTextField jFormattedTextField25;
private javax.swing.JFormattedTextField jFormattedTextField3;
private javax.swing.JFormattedTextField jFormattedTextField4;
private javax.swing.JFormattedTextField jFormattedTextField5;
private javax.swing.JFormattedTextField jFormattedTextField6;
private javax.swing.JFormattedTextField jFormattedTextField7;
private javax.swing.JFormattedTextField jFormattedTextField8;
private javax.swing.JFormattedTextField jFormattedTextField9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
// End of variables declaration
}


/*
 * @author haris_alkwitang
 */
public class SplashFrame extends javax.swing.JFrame {

    /**
     * Creates new form SplashFrame
     */
    public SplashFrame() {
        initComponents();
        setLabel();
        if (!Function.getResources("application.picture.splash").equals("")) {
            Function.setIcon(Function.getResources("application.picture.splash"), jLabel4);
        }
    }

    private void setLabel() {
        jButton1.setText(Function.getResources("button.enter"));
        jButton2.setText(Function.getResources("button.out"));
        jLabel1.setText(Function.getResources("label.firstline"));
        jLabel2.setText(Function.getResources("label.secondline"));
        jLabel3.setText(Function.getResources("label.thirdline"));
    }

    /**
     * This method is called from within the constructor to
     * initialize the form.
     *
     * WARNING: Do NOT modify this code. The content of this
     * method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setResizable(false);
        setUndecorated(true);
        jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
                )
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
                .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14));
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("SISTEM PENUNJANG KEPUTUSAN
KELAYAKAN");
        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1)
                    .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1)
                    .addContainerGap())
        );
    }
}


```

## 26. SplashFrame.java

```

/*
 * SplashFrame.java
 * Created on Feb 14, 2011, 9:00:15 AM
 */

package org.uin.spk.screen;
import org.uin.spk.util.Function;

```

```
jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("PEMBIAYAAN USAHA KECIL
MENEGAH");
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("BANK PEMBIAYAAN RAKYAR
SYARIAH WAKALUMI");
javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel2,
                javax.swing.GroupLayout.DEFAULT_SIZE, 415,
                Short.MAX_VALUE)
            .addComponent(jLabel3,
                javax.swing.GroupLayout.DEFAULT_SIZE, 415,
                Short.MAX_VALUE))
        .addGap(10, 10, 10)
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 24,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 24,
                        javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(10, 10, 10)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3))
                .addGap(36, 36, 36)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)))
            .addGap(66, 66, 66)
        );
);
jButton1.setFont(new java.awt.Font("Tahoma", 1, 14));
jButton1.setText("Enter");
jButton1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton1ActionPerformed(evt);
    }
});
jButton2.setFont(new java.awt.Font("Tahoma", 1, 14));
jButton2.setText("Out");
jButton2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton2ActionPerformed(evt);
    }
});
javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jPanel1,
                javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(10, 10, 10)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jPanel2,
                        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel1,
                        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(10, 10, 10)
            );
);
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    .addGroup(layout.createSequentialGroup()
        .addGap(84, 84, 84)
        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 154,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(119, 119, 119)
        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 154,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(72, 72, 72)))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
            .addComponent(jPanel2,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(jPanel1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
);
pack();
}// </editor-fold>

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    LoginFrame loginFrame = new LoginFrame();
    loginFrame.setLocationRelativeTo(null);
    loginFrame.setVisible(true);
    this.dispose();
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
// End of variables declaration
}
```

**27. Config.properties**

```

prefix.pk.pemohon=NSP
prefix.pk.kriteria.variable=KVP
prefix.pk.kriteria.nilai=KNP
prefix.pk.master.perhitungan=MPP

# Konfigurasi Koneksi Database
# Sesuaikan dengan environment server
db.ipdb=localhost
db.dbport=3306
db dbname=fuzzydb
db.user=root
db.pass=

# Start - Don't change this
db.url=jdbc:mysql://{ip}:{port}/{dbname}
db.driver=com.mysql.jdbc.Driver
# End

#Start Key
key.parent.jaminan=KVP001
key.parent.karakter=KVP002
key.parent.kemampuan=KVP003
key.parent.usaha=KVP004

key.child.nilaijaminan=KNP003
key.child.penghasilan=KNP007
key.child.pengeluaran=KNP011
key.child.kepemilikanempattinggal=KNP015
key.child.kondisitempattinggal=KNP020
key.child.jumlahtanggungan=KNP024
key.child.modalinvestasi=KNP028
key.child.omzet=KNP032
key.child.lokasiusaha=KNP036
key.child.jenistempatusaha=KNP040
key.child.lamausaha=KNP046
key.child.persaingan=KNP050
key.child.kepemilikanempatatusaha=KNP054
key.child.jumlahtenagakerja=KNP059
key.child.perilaku=KNP064
key.child.penampilan=KNP068
key.child.interaksi=KNP072
key.child.ketaatan=KNP076
#End Key

#####
#####START-
LANGUAGE#####
# Application Config
application.title=SPK Kelayakan Application
application.picture.login=gambar\loginorangkey.jpg
application.picture.splash=gambar\intro.jpg
application.picture.about.banner=gambar\banner.jpg
#application.picture.about.pic=pic/test4.jpg
application.picture.about.pic=gambar\Pas foto haris1.jpg

#application.numberformat=#,###.00

# Main Menu
#di tulis filenya full path
file.mainmenu.admin.absolutepath=D:\\Documents and
Settings\\haris\\My Documents\\NetBeansProjects\\SPK-
Pembayaran_UKM\\mainmenu2.html
file.mainmenu.manager.absolutepath=D:\\Documents and
Settings\\haris\\My Documents\\NetBeansProjects\\SPK-
Pembayaran_UKM\\mainmenu.html
# Report Name
application.report.name=printreport.jrxml

# Report Parameter
report.city=Jakarta,
report.jabatan=Manager
report.dateformat=dd-MM-yyyy

# Start - Access Control List
# 1 mean have access
# 0 mean dont have access

application.admin=admin
application.admin.nilaikriteria=0
application.admin.kriteria=0
application.admin.pemohon=1
application.admin.pemeriksaan=0
application.admin.perhitungan=0
application.admin.laporan=0

application.manager=manager
application.manager.nilaikriteria=1
application.manager.kriteria=1
application.manager.pemohon=0
application.manager.pemeriksaan=1
application.manager.perhitungan=1
application.manager.laporan=1

# End - Access Control List

# Start - System Rule
system.file.rule=rule.txt
system.file.separator=[=]
system.file.secondseparator=,
system.file.sl=SL
system.file.lrr=LRR
system.file.lrt=LRT
system.file.kl=K/TL
system.file.stl=STL

system.file.sb=SB
system.file.cb=CB
system.file.tb=TB

system.sl=Sangat Layak Untuk Dipinjamkan
system.lrr=Layak Dengan Resiko Rendah Untuk
Dipinjamkan
system.lrt=Layak Dengan Resiko Tinggi Untuk Dipinjamkan
system.kl=Kurang Layak Untuk Dipinjamkan
system.stl=Sangat Tidak Layak Untuk Dipinjamkan
# End - System Rule

# Menu Atas
menu.file=File
menu.kriteria=Kriteria
menu.pemohon=Pemohon
menu.pemeriksaan=Pemeriksaan
menu.about=About
menu.laporan=Laporan
menu.item.kriteria=Kriteria
menu.item.nilaikriteria=Nilai Kriteria
menu.item.pemohon=Pemohon
menu.item.pemeriksaan=Pemeriksaan
menu.item.perhitungan=Perhitungan
menu.item.mainmenu=Main Menu
menu.item.exit=Exit
menu.item.about=About
menu.item.laporan=Laporan

# Alert Language
alert.config.notfound=File config.properties tidak di temukan
alert.login.failedlogin=Failed to login
alert.db.failedlogin=Failed to connect to database server,
Please check your config!
alert.text.isi=Mohon diisi semua data-datanya.
alert.delete.pemohon=Are you sure to delete pemohon data..?
alert.isi.periksa=Mohon pilih semua kriterianya..
alert.delete.perhitungan=Are you sure to delete perhitungan
data..?

# Splash Screen
label.firstline=SISTEM PENUNJANG KEPUTUSAN
KELAYAKAN
label.secondline=PEMBIAYAAN USAHA KECIL
MENENGAH
label.thirdline=BANK PEMBIAYAAN RAKYAT
SYARIAH WAKALUMI

```

```

# Sistem Penunjang Keputusan Pembiayaan BPRS Wakalumi
label.about.1line=SPK Pembiayaan BPRS Wakalumi
label.about.2line=Mohamad Haris
label.about.3line=106093003109
label.about.4line=Sistem Informasi
label.about.5line=Mohamadhrs@gmail.com

# Header Table
table.header.idkriteria=Id Kriteria
table.header.kriteriainduk=Kriteria Induk
table.header.kriteriavariabel=Kriteria Variabel
table.header.parameter=Parameter
table.header.bobot=Bobot

table.header.nopemohon=No Pemohon
table.header.jenispemohon=Jenis Pemohon
table.header.namapemohon=Nama Pemohon
table.header.alamatpemohon=Alamat

table.header.idperhitungan=ID Perhitungan
table.header.datehitung=Tanggal Perhitungan

table.header.nilaipinjaman=Nilai Pinjaman
table.header.nilaihasil=Hasil Perhitungan
table.header.rangehasil=Rekomendasi

# Label Text
label.idkriteria=Id Kriteria:
label.kriteriainduk=Kriteria Induk :
label.idvariable=Id Variabel:
label.parameter=Parameter:
label.bobot=Bobot:
label.login.header=Isilah kolom dibawah ini dengan Benar.
label.login2.header=Assalamualaikum Wr. Wb.
label.username=Username
label.password=Password
label.nama>Nama
label.nopemohon=No Pemohon
label.jenispemohon=Jenis Pemohon
label.lama=Lama
label.baru=Baru
label.fotopemohon=Foto Pemohon
label.alamatrumah=Alamat Rumah
label.pekerjaan=Pekerjaan / Usaha
label.rencanapembiayaan=Rencana Pinjaman Pembiayaan
label.nilaipinjaman=Nilai Pinjaman
label.jangkawaktupembiayaan=Jangka Waktu Pembiayaan
label.jenisbarangjaminan=Jenis Barang Jaminan
label.nilaitaksasijaminan=Nilai Taksasi Jaminan
label.penyerahanadministrasi=Penyerahan Administrasi
label.memenuhi=Memenuhi
label.tidakmemenuhi=Tidak Memenuhi
label.rp=Rp
label.calon=Calon
label.nilaijaminan=Nilai Jaminan
label.penghasilanbulan=Penghasilan /bulan
label.pengeluaranbulan=Pengeluaran /bulan
label.kepemilikantempattinggal=Kepemilikan Tempat Tinggal
label.kondisitempattinggal=Kondisi Tempat Tinggal
label.tanggungan=Jumlah Tanggungan Keluarga
label.modal=Modal Investasi
label.omzet=Omzet /bln
label.lokasiusaha=Lokasi Usaha
label.jenistempatusaha=Jenis Tempat Usaha
label.lamausaha=Lama Usaha
label.persaingan=Persaingan
label.kepemilikantempatusaha=Kepemilikan Tempat Usaha
label.jumlantenagakerja=Jumlah Tenaga Kerja
label.perilakusehari=Perilaku sehari-hari
label.penampilan=Penampilan
label.interaksimasyarakat=Interaksi Masyarakat
label.ketaatanberibadah=Ketaatan Beribadah
label.jaminan=Jaminan
label.kemampuan=Kemampuan

label.usaha=Usaha
label.karakter=Karakter
label.date=Date
label.jumlahjaminan=Jumlah Jaminan
label.jumlahusaha=Jumlah Usaha
label.jumlahkemampuan=Jumlah Kemampuan
label.jumlahkarakter=Jumlah Karakter
label.hasilpenilaian=Hasil Penilaian
label.rekomendasi=Rekomendasi Penilaian
label.jumlahpinjamandiberikan=Jumlah Pinjaman Yang akan dipinjamkan
label.hasilfuzzy=Hasil Perhitungan Fuzzy
label.namapemohon=Nama Pemohon
label.rekomendasi2=Rekomendasi

# Button Text
button.new>New
button.save=Save
button.delete=Delete
button.refresh=Refresh
button.login=Login
button.cancel=Cancel
button.enter=Enter
button.out=Out
button.pilihfile=Pilih File
button.preview=Preview
button.periksa=Periksa
button.reset=Reset
button.pilihcalon=Pilih Calon
#####
END-LANGUAGE#####

```

## **WAWANCARA**

Responden : Ibu Silvy Sy.

Pewawancara : Mohamad Haris

Tujuan wawancara : Mengetahui prosedur pemberian dan kriteria atau parameter yang berlaku.

Poin Wawancara:

1. Assalamualaikum Wr.Wb, Semoga kabar ibu baik, sehat dan sejahtera selalu?

Wa'alaikumussalam. aaamin

2. Maaf mengganggu ibu yang saya hormat, saya M.Haris Mahasiswa UIN jakarta, Saya ingin mengajukan beberapa pertanyaan yang berkaitan dengan pemberian pemberian Di BPRS Wakalumi?

Ya, Silahkan ....!

3. Sebelumnya saya boleh tahu profil BPR Syariah WAKALUMI ini bu?

Oh iya, tentu .. kamu bisa liat di arsip, dan nanti saya berikan softcopinya.

4. Saya ingin tahu, tugas atau pelayanan apa saja yang ada di BPRS Wakalumi untuk membantu dalam Pemberian nasabah dan Masyarakat sekitar?

Pelayanan yang ada di BPRS ini seperti tabungan, pemberian untuk usaha, modal kerja, pembelian atau penyewaan tempat usaha, pembelian peralatan usaha baru, pemberian motor, renovasi rumah. Dengan bentuk-bentuk pemberian syariah seperti Murahabah, Mudharabah, Musyarakah, Ijarah, Multijasa, Qorhaul Hasan dan Rahn

5. Usaha apa sajakah yang pernah dan sering mengajukan pemberian Ke BPRS ini?

Biasanya yang mengajukan pemberian disini seperti pedagang sembako, daging, kelontong, industri rumah tangga dan sebagianya.

6. Berapa maksimal dan minimal pinjaman yang diberikan di BPRS, serta berapa besar bunganya?

Pinjaman minimal Rp. 5.000.000 sedangkan maksimalnya Rp.350.000.000 dan BPRS ini tidak menggunakan bunga akan tetapi bagi hasil dari usaha yang di sepakati. Untuk simulasi gambaran nya nanti saya kasih tabel angsuran nya

7. Bagaimana proses pemberian pembiayaan di BPRS Wakalumi ini/ Tahap apa saja untuk pengajuan kredit/pembiayaan?

Pada dasarnya proses pemberian pembiayaan yang ada disini hampir sama dengan lembaga keuangan lainnya. Untuk memperoleh pinjaman, calon debitur harus mengisi formulir pemohonan pembiayaan dan melengkapi persyaratan yang ada seperti foto copy KTP Suami dan Istri, foto copy kartu keluarga, foto copysurat nikah, peta lokasi rumah, surat keterangan sebagai karyawan, copy slip gaji terakhir, data keuangan usaha 3 bulan terakhir, rekening koran, foto copy ijazah, BPKB kendaraan bermotor, akta jual beli atas nama sendiri, sertifikat hak milik atas nama sendiri dengan ketentuan disini. Jika syaratnya memenuhi, pihak kami Account Manajer (petugas pembiayaan) ke calon debitur untuk melakukan survey. AM mencocokkan kebenaran data calon debitur, lalu AM juga menilai kelayakan dengan melihat seluruh parameter atau kriteria yang dapat dijadikan tolak ukur penilaian, diantaranya parameter kinerja nasabah dan usahanya. Lalu kami menganalisa data hasil surveynya. Jika calon debitur layak menerima pembiayaan maka pinjaman diberikan, jika calon debitur tidak layak nanti ada informasi alasan penolakan

8. Sistem Apa yang dipakai untuk menganalisa kelayakan pembiayaan di BPR Syariah ini?

Oh.... kalau untuk itu kita pakai teknik pertimbangan bisa disebut juga scoring...! maksudnya semua kriteria yang mempengaruhi diberi skor, hasil keseluruhan mengidentifikasi layak atau tidaknya calon debitur memperoleh pembiayaan. Jadi kita punya kriteria, dimana masing-masing kriteria memiliki skor atau nilai yang yang berbeda-beda. Skor-skor itulah yang nanti akan menentukan kelayakan tidaknya calon debitur memperoleh pinjaman.

9. Kriteria-kriteria apa aja ibu?

Ada 4 kriteria yaitu pertama Jaminan, kedua Usaha , ketiga Kemampuan dan keempat Karakter. Setiap kriteria terdiri dari beberapa parameter, untuk lebih jelasnya ada di lampiran.

10. Terimakasih atas penjelasannya. Kalau saya masih butuh data, saya bisa bertemu ibu lagi?

Sama-sama ris...! ya nanti kalo mau tanya-tanya lagi sebelumnya hubungi saya dulu ya...!

11. Assalamualaikum.. bu?

Wa'alaikumussalam Warahmatullahi Wa Barakaatuh..!