

**KOMPARASI ALGORITMA C4.5, SVM, NEURAL NETWORK  
BERBASIS PARTICLE SWARM OPTIMAZATION UNTUK  
PENENTUAN KELAYAKAN KREDIT**



**TESIS**

**SYAIFUR RAHMATULLAH A.R**

**1400529**

**PROGRAM PASCASARJANA MAGISTER ILMU KOMPUTER  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
NUSA MANDIRI**

JAKARTA  
2015

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini :

Nama : Syaifur Rahmatullah Abdul Rozak

NIM : 14000529

Program Studi: Magister Ilmu Komputer

Jenjang : Strata Dua (S2)

Konsentrasi : *E-Business*

Dengan ini menyatakan bahwa tesis yang telah saya buat dengan judul: “Komparasi Algoritma C4.5, SVM, Neural Network berbasis Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit” adalah hasil karya sendiri, dan semua sumber baik yang kutip maupun yang dirujuk telah saya nyatakan dengan benar dan tesis belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tesis yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri dicabut/dibatalkan.

Jakarta, 1 Maret 2015

Yang Menyatakan,

Syaifur Rahmatullah Abdul Rozak

## HALAMAN PENGESAHAN

Tesis ini diajukan oleh :

Nama : Syaifur Rahmatullah A.R

NIM : 14000529

Program Studi : Magister Ilmu Komputer

Jenjang : Strata Dua (S2)

Konsentrasi : *E-Business*

Judul Tesis : Komparasi Algoritma C4.5, SVM, NN berbasis  
Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit

Telah Berhasil Dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Ilmu Komputer (M.Kom) pada Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri).

Jakarta, 11 Maret 2015

Pascasarjana Magister Ilmu Komputer

STMIK Nusa Mandiri

Direktur



Prof. Dr. Ir Kaman Nainggolan MS

## DEWAN PENGUJI

Penguji I : Dr. Sularso Budilaksono

  
.....

Penguji II : Dr. Windu Gata, M.Kom

  
.....

Penguji III / Pembimbing : Dr. Dana Indra Sensuse, MILS, Ph.D .....



## **KATA PENGANTAR**

Puji Syukur alhamdulillah, Penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan Rahmat dan Karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan tesis ini dengan lancar. Adapun judul penulisan skripsi yang penulis ambil adalah sebagai berikut “ Komparasi Algoritma C4,5, SVM, Neural Network berbasis Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit” Adapun tujuan dari penulisan tesis ini dibuat sebagai salah satu syarat untuk mendapatkan gelar Magister Ilmu Komputer (M.Kom) pada Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa mandiri). Tesis ini dibuat berdasarkan hasil penelitian mengenai Komparasi Algoritma C4,5, SVM, Neural Network berbasis Particle Swarm Optimazation untuk menentukan algoritma yang paling akurat dalam penentuan kelayakan kredit. Penulis juga mencari dan menganalisa berbagai macam sumber referensi, baik dalam bentuk Jurnal Ilmiah, Buku – buku Literatur, Internet, dan lain – lain yang terkait dengan pembahasan pada Tesis ini.

Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan skripsi ini tidak akan berjalan lancar. Oleh karena itu pada kesempatan ini, izinkanlah penulis menyampaikan ucapan terima kasih kepada :

1. Bapak Dr. Dana Indra Sensuse, Selaku Pembimbing Tesis yang telah menyediakan waktu, pemikiran dan tenaga dalam membimbing penulis dalam menyelesaikan tesis ini.

2. Keluarga yang telah memberikan dukungan Moral Kepada Penulis.
3. Seluruh Staf Pengajar (Dosen) Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri yang telah memberikan pelajaran yang berarti bagi penulis selama menempuh studi
4. Seluruh Rekan Kerja penulis di Bina Sarana Informatika yang telah banyak memberikan masukan – masukan yang bermanfaat pada penelitian ini, khususnya Staff bagian Marketing Komunikasi BSI
5. Rekan – Rekan Seperjuangan.

serta teman – teman yang terlalu banyak untuk disebut satu persatu, terima kasih atas do'a dan dukungannya. Penulis menyadari bahwa dalam penulisan skripsi ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan dimasa yang akan datang.

Akhir kata semoga tesis ini dapat bermanfaat dan menambah wawasan bagi penulis khususnya dan bagi para pembaca serta pihak yang berkepentingan. Semoga Allah SWT memberikan pahalanya kepada semua pihak yang telah memberikan bantuan dan bimbingannya.

Jakarta, Maret 2015

Penulis

**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH**  
**UNTUK KEPENTINGAN AKADEMIS**

Yang bertanda tangan dibawah ini :

Nama : Syaifur Rahmatullah Abdul Rozak  
NIM : 14000529  
Program Studi : Magister Ilmu Komputer  
Jenjang : Strata Dua (S2)  
Konsentrasi : *E-Business*  
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan ijin kepada pihak Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri) Hak Bebas Royalti Non-Eksklusif (Non-exclusive Royalti-Free Right) atas karya ilmiah kami yang berjudul : “Komparasi Algoritma C4,5, SVM, Neural Network berbasis Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit” beserta perangkat yang diperlukan (apabila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini pihak STMIK Nusa Mandiri berhak menyimpan, mengalih media atau bentuk-kan, mengelolaannya dalam pangkalan data (database), mendistribusikannya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut. Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak STMIK Nusa Mandiri, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini. Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 1 Maret 2015

Yang Menyatakan

( Syaifur Rahmatullah Abdul Rozak )

## ABSTRAK

Nama : Syaifur Rahmatullah Abdul Rozak  
NIM : 14000529  
Program Studi: Magister Ilmu Komputer  
Jenjang : Strata Dua (S2)  
Konsentrasi : *E-Business*  
Judul : “Komparasi Algoritma C4.5, SVM, Neural Network berbasis Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit”

Kredit macet merupakan salah satu resiko kredit yang dihadapi oleh pelaku industri keuangan dan perbankan. Kredit macet terjadi apabila dalam jangka panjang, lembaga keuangan atau perbankan tidak dapat menarik pinjaman kredit dalam waktu yang telah ditentukan. Kredit macet memiliki dampak yang buruk bagi penyedia kredit yaitu berupa resiko kerugian. Tentu saja hal ini tidak boleh dibiarkan berlarut-larut dan harus dicari jalan keluarnya. Namun untuk menjamin akurasi dalam penentuan kelayakan kredit diperlukan suatu algoritma yang akurat. Oleh karena itu perlu diadakan sebuah penelitian yang bertujuan untuk mencari algoritma yang akurat dengan cara mengkomparasi beberapa algoritma diantaranya C4.5, SVM dan Neural Network. Untuk lebih meningkatkan akurasi ketiga algoritma tersebut di optimasi dengan Particle Swarm Optimazation. Hasil penelitian berupa confusion matrix membuktikan akurasi Neural Network berbasis Particle Swarm Optimazation memiliki tingkat akurasi sebesar 96,67% dan hasil AUC sebesar 0.965. Walaupun hasil AUC tertinggi ada pada algoritma SVM berbasis PSO yaitu 0,989 tetapi algoritma Neural Network berbasis Particle Swarm Optimazation merupakan algoritma yang paling akurat dikarenakan lebih dominan dalam uji T-test dari algoritma lainnya.

**Kata Kunci** : Data Mining, *Neural Network*, SVM, C4.5, *Particle Swarm Optimazation*

## **ABSTRACT**

*Name* : Syaifur Rahmatullah Abdul Rozak  
*NIM* : 14000529  
*Study Of Program* : Magister Ilmu Komputer  
*Levels* : Strata Dua (S2)  
*Concentration* : E-Business  
*Title* : “Comparison Algorithm C4.5, SVM, Neural Network-based Particle Swarm Optimazation For Credit Eligibility Determination”

*Bad credit is one of the credit risk faced by the financial and banking industry. Bad credit occurs when the long-term, financial institutions or banks can not attract loans within the allotted time. Bad debts have impacted the credit provider in the form of risk of loss Of course this should not be allowed to drag on and must find a way out. However, to ensure accuracy in determining credit worthiness required an accurate algorithm. Therefore, there should be a study that aims to find an algorithm that accurately by means of some algorithms such mengkomparasi C4.5, SVM and Neural Network. To further improve the accuracy of the three algorithms in Particle Swarm Optimization with Optimazation. Confusion matrix berupan research results prove the accuracy of Neural Network-based Particle Swarm Optimazation possess accuracy rate of 96.67% and the AUC of 0965. Although the highest AUC results SVM algorithm based on PSO is 0.989 but Neural Network algorithm based Particle Swarm Optimazation is the most accurate algorithm due to the more dominant in the T-test trials of other algorithms.*

**Key Word** : Data Mining, Neural Network, SVM, C4.5, Particle Swarm Optimazation

# DAFTAR ISI

	Halaman
Halaman Sampul .....	i
Halaman Judul .....	ii
Halaman Pernyataan Orisinalitas.....	iii
Halaman Pengesahan .....	iv
Lembar Konsultasi.....	v
Kata Pengantar .....	vii
Abstrak .....	ix
Abstract .....	x
Daftar Isi .....	xi
Daftar Tabel .....	xiii
Daftar Gambar .....	xiv
Daftar Lampiran.....	xv
<b>BAB I    Pendahuluan .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah.....	4
1.3. Rumusan Masalah .....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian .....	4
1.6. Ruang Lingkup Penelitian.....	5
1.7. Hipotesa.....	5
1.8. Sistematika Penulisan .....	5

<b>BAB II</b>	<b>Landasan/Kerangka Pemikiran.....</b>	<b>7</b>
2.1.	Tinjauan Pustaka .....	7
2.2.	Tinjauan Studi .....	38
2.3.	Tinjauan Perusahaan .....	41
2.4.	Kerangka Pemikiran .....	44
<b>BAB III</b>	<b>Metode Penelitian .....</b>	<b>45</b>
3.1.	Desain Penelitian .....	45
3.2.	Data Nasabah .....	46
3.3.	Data <i>Preparation</i> .....	46
3.4.	Pengujian Algoritma .....	50
3.5.	Evaluasi Hasil Pengujian.....	51
3.6.	Pengujian <i>T-test</i> .....	52
3.7.	Algoritma Terbaik.....	52
3.8.	Perancangan <i>Prototype</i> .....	52
3.9.	Pengujian <i>Prototype</i> .....	52
<b>BAB IV</b>	<b>Hasil Penelitian dan Pembahasan .....</b>	<b>53</b>
4.1.	Pengolahan Data .....	53
4.2.	Pembahasan.....	53
4.3.	Hasil Penelitian .....	81
4.4.	Perancangan dan Pengujian <i>Prototype</i> .....	93
4.5.	Implikasi Penelitian.....	96
<b>BAB V</b>	<b>Kesimpulan dan Saran.....</b>	<b>98</b>
5.1.	Kesimpulan .....	98
5.2.	Saran .....	98
	Daftar Pustaka .....	100
	Lampiran .....	104

## DAFTAR TABEL

Halaman

Tabel 2.1 AND Problem .....	16
Tabel 2.2 Input nilai bias, bobot dan bias .....	25
Tabel 2.3 Perhitungan untuk bias dan bobot baru .....	27
Tabel 2.4 Confusion Matrix .....	32
Tabel 2.5 Penelitian Sebelumnya yang Terkait .....	39
Tabel 3.1 Kategori Atribut .....	47
Tabel 3.2 Data Hasil <i>Transformation Min Max</i> .....	49
Tabel 3.3 Sampel Data Training .....	49
Tabel 3.4 Spesifikasi Hardware dan Software .....	51
Tabel 4.1 Tabel Informasi Gain C4.5 .....	54
Tabel 4.2 Bobot Atribut dengan Metode <i>decision tree</i> dengan PSO.....	59
Tabel 4.3 Tabel Nilai Input Awal .....	62
Tabel 4.4 Tabel Nilai Bobot Awal.....	62
Tabel 4.5 Tabel Nilai Bias Awal .....	62
Tabel 4.6 Tabel Penelitian Penentuan Training Cycles .....	64
Tabel 4.7 Tabel Penelitian Penentuan Nilai <i>Learning Rate</i> .....	65
Tabel 4.8 Tabel Penelitian Penentuan Nilai Momentum .....	65
Tabel 4.9 Tabel penentuan <i>Neuron Size</i> .....	66
Tabel 4.10 Tabel Nilai bobot <i>hidden layer</i> .....	68
Tabel 4.11 Tabel Nilai Bobot Akhir untuk output layer .....	70
Tabel 4.12 Bobot Atribut dengan Metode <i>Neural Network</i> dengan PSO.....	71
Tabel 4.13 Percobaan Penentuan Nilai <i>Training Cycles NN</i> berbasis PSO.....	73
Tabel 4.14 Percobaan Penentuan Nilai <i>Learning Rate NN</i> berbasis PSO .....	73
Tabel 4.15 Percobaan Penentuan Nilai <i>Nilai Momentum NN</i> berbasis PSO.....	74
Tabel 4.16 Percobaan Penentuan Nilai <i>Neuron Size NN</i> berbasis PSO.....	75
Tabel 4.17 Hasil Eksperimen <i>Support Vector Machine</i> .....	79
Tabel 4.18 Bobot Atribut dengan Metode <i>SVM</i> dengan PSO .....	75
Tabel 4.19 Hasil Eksperimen <i>SVM</i> berbasis PSO .....	80
Tabel 4.20 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> C4.5 .....	82
Tabel 4.21 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> C4.5 PSO.....	83
Tabel 4.22 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> SVM.....	84
Tabel 4.23 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> SVM PSO.....	85
Tabel 4.24 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> NN.....	86
Tabel 4.25 Nilai <i>Accuracy, Sensitivity, Specificity, ppv, dan npv</i> NN PSO.....	87
Tabel 4.26 Perbandingan <i>Performance</i> Metode.....	91
Tabel 4.27 Uji Statistik T-Test .....	91
Tabel 4.28 Hasil Perbandingan Seluruh Pengujian .....	93
Tabel 4.29 Data Testing .....	95

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Proses Data Mining menurut CRISP-DM .....	8
Gambar 2.2 Konsep SVM untuk mencari <i>hyperlane</i> terbaik.....	14
Gambar 2.3 Contoh konsep pohon keputusan sederhana.....	17
Gambar 2.4 Arsitektur <i>Artificial Neural Network</i> .....	20
Gambar 2.5 Contoh Arsitektur <i>Neural Network Multilayer feed- forward</i> .....	25
Gambar 2.6 Struktur dasar PSO .....	30
Gambar 2.7 Grafik ROC .....	34
Gambar 2.8 Struktur Organisasi Bank DKI .....	43
Gambar 2.9 Kerangka Pemikiran .....	44
Gambar 3.1 Model yang Di usulkan .....	45
Gambar 3.2 Proses <i>Replace Missing Value</i> Pada Rapidminer .....	46
Gambar 3.3 Hasil Proses <i>Replace Missing Value</i> Pada Rapidminer .....	47
Gambar 3.4 Proses <i>Remove Duplicate</i> Pada Rapidminer .....	49
Gambar 4.1 <i>K-fold Cross Validation</i> Algoritma C4.5.....	55
Gambar 4.2 <i>Decision Tree</i> C4.5 .....	56
Gambar 4.3 <i>K-fold Cross Validation</i> Algoritma C4.5 berbasis PSO .....	58
Gambar 4.4 <i>Decision Tree</i> C4.5 berbasis PSO.....	59
Gambar 4.5 Desain model algoritma <i>neural network</i> .....	61
Gambar 4.6 Arsitektur model algoritma <i>neural network</i> .....	67
Gambar 4.7 Arsitektur model algoritma <i>neural network</i> .....	68
Gambar 4.8 Model Algoritma NN berbasis PSO .....	71
Gambar 4.9 Desain model algoritma <i>support vector machine</i> .....	76
Gambar 4.10 Desain model algoritma <i>support vector machine</i> berbasis PSO .....	78
Gambar 4.11 Hasil Akurasi Algoritma C4.5.....	81
Gambar 4.12 Hasil Akurasi Algoritma C4.5 berbasis PSO .....	82
Gambar 4.13 Hasil Akurasi Algoritma SVM.....	83
Gambar 4.14 Hasil Akurasi Algoritma SVM berbasis PSO .....	84
Gambar 4.15 Hasil Akurasi Algoritma NN.....	85
Gambar 4.16 Hasil Akurasi Algoritma NN berbasis PSO .....	86
Gambar 4.17 Nilai AUC C4.5 dalam <i>ROC Curve</i> .....	88
Gambar 4.18 Nilai AUC C4.5 berbasis PSO dalam <i>ROC Curve</i> .....	88
Gambar 4.19 Nilai AUC SVM dalam <i>ROC Curve</i> .....	89
Gambar 4.20 Nilai AUC SVM berbasis PSO dalam <i>ROC Curve</i> .....	89
Gambar 4.21 Nilai AUC NN dalam <i>ROC Curve</i> .....	90
Gambar 4.22 Nilai AUC NN berbasis PSO dalam <i>ROC Curve</i> .....	90
Gambar 4.23 Model T-Test pada <i>Decision Tree</i> dan <i>Neural Network PSO</i> .....	92
Gambar 4.24 Hasil Model T-Test pada C4.5 dan <i>Neural Network PSO</i> .....	92
Gambar 4.25 Protoype Aplikasi Penentuan Kelayakan Kredit .....	94
Gambar 4.26 Grafik Hasil Kuisisioner .....	95
Gambar 4.25 Grafik Hasil Uji Coba <i>Prototype</i> .....	96

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Keterangan Riset.....	103
Lampiran 2 Contoh Kuisioner ISO 9126 .....	104
Lampiran 3. Dataset Nasabah Kredit (Data Asli).....	105
Lampiran 3. Dataset Nasabah Kredit (Transform) .....	106



**LEMBAR KONSULTASI BIMBINGAN  
TESIS**

**Pascasarjana Magister Ilmu Komputer  
STMIK Nusa Mandiri**

Nama : Syaifur Rahmatullah Abdul Rozak  
NIM : 14000529  
Dosen Pembimbing : Dr. Dana Indra Sensuse  
Judul Tesis : Komparasi Algoritma C4.5, SVM, Neural Network  
Berbasis Particle Swarm Optimazation untuk  
Penentuan Kelayakan Kredit.



No	Tanggal Bimbingan	Materi Bimbingan	Paraf dosen Pembimbing
1	7 November 2014	Pengajuan Judul	
2	16 November 2014	Revisi Judul dan Pengajuan Bab1	
3	14 Desember 2014	Revisi Bab 1 dan Pengajuan Bab II	
4	11 Januari 2015	Revisi Bab II	
5	18 Januari 2015	Revisi Bab II dan Pengajuan Bab III	
6	22 Februari 2015	Revisi Bab III dan Pengajuan Bab IV, Bab V	
7	1 Maret 2015	Acc Keseluruhan	

Bimbingan dimulai pada tanggal : 7 November 2014  
Bimbingan diakhiri pada tanggal : 1 Maret 2015  
Jumlah pertemuan : 7

Jakarta, 1 Maret 2015

Dosen Pembimbing



( Dr. Dana Indra Sensuse )

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Penulisan**

Kredit adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga. Sedangkan Kredit Tanpa Jaminan atau Kredit Tanpa Agunan (KTA) adalah kredit yang diberikan bank atau lembaga keuangan dalam bentuk uang tunai, yang dapat diperoleh tanpa memberikan jaminan. Kemudahan ini banyak dimanfaatkan oleh para nasabah untuk meminjam sejumlah dana dari bank & koperasi untuk digunakan berbagai keperluan nasabah (UU Perbankan No. 10 tahun 1998).

Penilaian kredit merupakan topik yang penting dalam pengelolaan resiko keuangan. Krisis keuangan global baru-baru ini memaksa pelaku industri keuangan dan perbankan untuk lebih berhati-hati dalam menetapkan kebijakan. Laporan bank Indonesia menunjukkan bahwa pada akhir tahun 2012 tercatat kredit macet senilai Rp. 33.401.000.000.000 naik hingga 17,64% dibandingkan dengan th 2011 sebesar Rp. 28.396.000.000.000 (Indonesia, Bank, 2013). Hal tersebut menunjukkan bahwa tingkat kredit macet mengalami kenaikan yang tajam. Sedangkan perkembangan jumlah debitur semakin bertambah yaitu 59.637 pada Desember 2012 dan meningkat menjadi 63.364 pada Mei 2013 (Indonesia, Bank, 2013).

Kredit macet merupakan salah satu resiko kredit yang dihadapi oleh pelaku industri keuangan dan perbankan. Kredit macet terjadi apabila dalam jangka panjang, lembaga keuangan atau perbankan tidak dapat menarik pinjaman kredit dalam waktu yang telah ditentukan (Jianguo & Tao, 2008). Kredit macet memiliki dampak yang buruk bagi penyedia kredit yaitu berupa resiko kerugian (Hian, Wei & Chwee, 2006). Tentu saja hal ini tidak boleh dibiarkan berlarut-larut dan harus dicari jalan keluarnya, karena masalah ini dapat mengakibatkan penghentian operasi dan mengarah kepada kebangkrutan lembaga penyedia kredit tersebut (Jianguo & Tao, 2008)

Kredit macet dapat dihindari dengan cara melakukan analisa kredit yang akurat terhadap calon debitur (Bellotti & Crook, 2007). Tujuan utamanya adalah untuk memperoleh keyakinan bahwa debitur akan dapat memenuhi setiap kewajibannya dan memastikan kualitas kredit tetap lancar sampai berakhirnya perjanjian kredit. Keakuratan penilaian kredit sangat penting untuk profitabilitas lembaga keuangan (Gang, Jinxing, Jian & Hongbing, 2011).

Penilaian kredit mengelompokkan para calon debitur menjadi dua jenis yaitu debitur baik dan debitur buruk. Debitur baik memiliki kemungkinan besar akan membayar kewajiban keuangannya dengan lancar, sedangkan debitur buruk memiliki kemungkinan besar mengalami kredit macet (Gang, Jinxing, Jian & Hongbing, 2011). Penilaian kredit sangat penting karena banyak keuntungan yang diperoleh yaitu mengurangi biaya analisa kredit, pengambilan keputusan lebih cepat, pemantauan lebih dekat dengan data yang ada dan memungkinkan untuk menetapkan calon debitur prioritas (Ping, 2009). Penilaian kredit juga bermanfaat bagi penyedia kredit untuk mengukur dan mengelola risiko keuangan dalam memberikan kredit sehingga mereka dapat membuat keputusan yang lebih baik, lebih cepat dan lebih obyektif (Hian, Wei & Chwee, 2006).

Penelitian terdahulu mengenai topik penilaian kredit telah banyak dilakukan seperti penelitian yang dilakukan diantaranya adalah dari peneliti Yi Jian yang mengambil judul *A Bank Customer Credit Evaluation Based on the Decision Tree and the Simulated Annealing Algorithm* yang bertujuan untuk penentuan nasabah baik dan buruk dalam pemberian kredit, dan terbukti algoritma C4.5 merupakan algoritma yang memiliki keandalan, efisiensi dan kesederhanaan dalam hal pengklasifikasian jenis nasabah (Yi Jian, Ya Chen, & Xiangjian Dia, 2007). Penelitian lain mengenai resiko pemberian kredit juga dilakukan oleh Ning liu dengan judul *Research and Application of PSO-BP Neural Network in Credit Risk Assessment* dengan cara membandingkan dua buah seleksi fitur pada satu algoritma yaitu *Genetic Algorithm BP Neural Network* dengan *PSO BP Neural Network*, dari hasil penelitiannya di simpulkan bahwa *PSO BP Neural Network* lebih baik dibandingkan *GA BP Neural Network*, dalam jurnal ini peneliti memilih *Neural Network* sebagai algoritma pemecahan masalah prediksi resiko kredit dikarenakan algoritma ini dapat melakukan pengolahan data dalam jumlah

besar, serta memiliki kemampuan untuk menangani informasi non linear (Ning Liu, En Jun Xia, Li Yang, 2010). Peneliti selanjutnya yang membahas tentang resiko pemberian kredit adalah Yun Li dengan judul penelitian *Application of the PSO - SVM Model for Credit Scoring*, penelitian kali ini menggunakan algoritma SVM dengan metode penelitian membandingkan beberapa fitur seleksi untuk meningkatkan akurasi SVM dan menghasilkan bahwa PSO dengan SVM memiliki akurasi yang tinggi dalam pemecahan masalah resiko penentuan kredit (Yun Li, Qiu Yancoo, Hua Zhang, 2011)

Seleksi fitur adalah langkah untuk memilih dan mendapatkan informasi yang lebih berharga dari data dengan fitur yang besar. Atribut dan informasi yang berlebihan yang dimasukkan kedalam model penilaian kredit mengakibatkan banyaknya waktu dan biaya yang dikorbankan bahkan akan mengurangi tingkat akurasi dan kompleksitas yang lebih tinggi. Untuk itu diperlukan metode seleksi atribut pada data set dengan jumlah atribut yang besar untuk meningkatkan hasil akurasi (Ping, 2009).

*Particle swarm optimization (PSO)* merupakan teknik komputasi evolusioner yang mampu menghasilkan solusi optimal secara global dalam ruang pencarian melalui interaksi individu dalam segerombolan partikel. Setiap partikel menyampaikan informasi berupa posisi terbaiknya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang terbaik tersebut (Shuzhou & Bo, 2011). *Particle swarm optimization* dapat digunakan sebagai teknik optimasi untuk mengoptimalkan subset fitur dan parameter secara bersamaan (Yun, Qiu-yan & Hua, 2011).

Pada kesempatan ini penelitian yang akan dilakukan adalah menganalisis komparasi algoritma klasifikasi *data mining*, diantaranya adalah algoritma, algoritma *Decision Tree*, algoritma *Neural Network*, dan algoritma *Support Vector Machine* serta penerapan metode *Particle swarm optimization (PSO)* dalam penyeleksian atribut dengan tujuan untuk mendapatkan algoritma yang paling akurat dalam penentuan kelayakan pemberian kredit.

## 1.2 Identifikasi Masalah

Identifikasi permasalahan pada penelitian ini adalah sebagai berikut:

- a. Belum diketahuinya metode menganalisis komparasi algoritma-algoritma klasifikasi *data mining* pada penentuan kelayakan pemberian kredit
- b. Belum diketahuinya algoritma klasifikasi *data mining* yang akurat pada penentuan kelayakan pemberian kredit.

## 1.3 Rumusan Masalah

Rumusan masalah yang berdasarkan kepada beberapa jurnal yang telah disebutkan diatas, *Neural Network* berbasis *Particle Swarm Optimazation* merupakan algoritma terbaik dikarenakan mempunyai nilai eror terkecil yaitu 1,6487%. Oleh karena itu dalam penelitian kali ini akan di komparasi ketiga Algoritma klasifikasi yaitu *Neural Network*, *Support Vector Machine*, dan *Decision tree*, dengan berbasis *Particle Swarm Optimazation* untuk mengetahui apakah algoritma *Neural Network* berbasis *Particle Swarm Optimazation* merupakan algoritma klasifikasi terbaik dalam hal klasifikasi kelayakan pemberian kredit.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendapatkan algoritma yang paling akurat dalam penentuan kelayakan pemberian kredit. Algoritma- algoritma yang digunakan adalah *Decision Trees*, *Neural Network* dan *Support Vector Machines* berbasis *Particle Swarm Optimazation*

## 1.5 Manfaat Penelitian

Manfaat dari penyusunan tesis ini adalah :

- a. Manfaat praktis dari hasil penelitian ini adalah dapat digunakan oleh pihak bank untuk menentukan kelayakan pemberian kredit kepada debitur dengan lebih baik.
- b. Manfaat kebijakan dari hasil penelitian ini adalah dapat digunakan sebagai bahan pertimbangan dalam pemberian kredit kepada debitur pada bank DKI Jakarta Cab, Bendungan Hilir.
- c. Manfaat teoritis dari penelitian ini yaitu dapat memberikan kontribusi pada metode *data mining* dalam membandingkan atau mengkomparasikan beberapa algoritma klasifikasi *data mining* untuk kasus-kasus yang belum

pernah dilakukan komparasi atau pada kasus-kasus yang sudah pernah dilakukan komparasi tetapi dengan algoritma yang berbeda.

### **1.6 Ruang Lingkup**

Data yang akan di analisis merupakan data nasabah kreditur yang di ambil dari Bank DKI Jakarta Cabang Bendungan Hilir dari April 2012 sampai dengan Juni 2013. Batasan permasalahan pada penelitian ini dibatasi pada bagaimana menganalisis komparasi model-model teknik klasifikasi *data mining* untuk algoritma *Decision Trees*, *Neural Network* dan *Support Vector Machines* berbasis *Particle Swarm Optimization* yang yang di terapkan terhadap kelayakan pemberian kredit pada bank DKI Jakarta Cabang Bendungan Hilir.

### **1.7 Hipotesa**

Dalam penelitian ini diduga Algoritma *Neural Network* berbasis *Particle Swarm Optimazation* merupakan algoritma paling akurat dalam penentuan kelayakan kredit.

### **1.8 Sistematika Penulisan**

Sistematika penulisan tesis ini terdiri dari 5 (lima) bab, dimana tiap bab terdiri dari beberapa sub bab yaitu :

#### **BAB I : PENDAHULUAN**

Bab ini menguraikan tentang latar belakang, masalah penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

#### **BAB II : LANDASAN TEORI DAN KERANGKA PEMIKIRAN**

Bab ini membahas tentang tinjauan pustaka, tinjauan studi, tinjauan obyek penelitian, kerangka konsep penelitian, dan hipotesis.

#### **BAB III : METODOLOGI PENELITIAN**

Bab ini membahas tentang metode penelitian, langkah-langkah penelitian, metode pemilihan sampel, metode pengumpulan data, instrumen penelitian, dan jadwal penelitian.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Membahas mengenai hasil penelitian, pengujian data dan implikasi penelitian yang berhubungan dengan masalah yang diteliti.

**BAB V : PENUTUP**

Memberikan kesimpulan dan saran penelitian.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Dalam penulisan tesis ini penulis melakukan tinjauan studi menggunakan buku dan jurnal yang berhubungan dengan tema yang dipilih. Secara lebih detail tinjauan studi dalam penulisan tesis ini dapat di jelaskan sebagai berikut:

##### 2.1.1 *Data Mining*

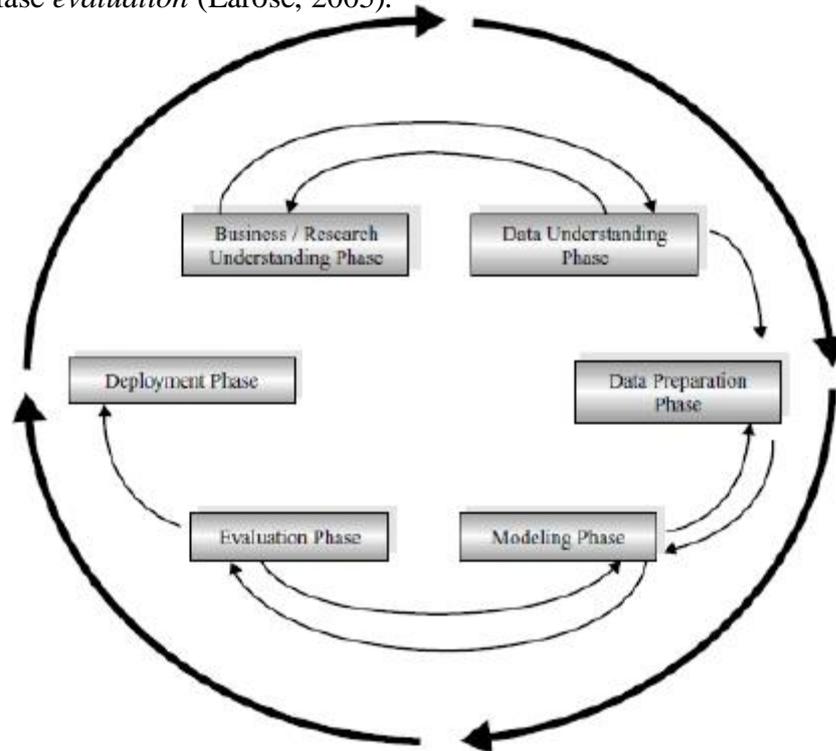
*Data mining* adalah aplikasi algoritma spesifik untuk mengekstrak pola dari data (Abraham, Grosan & Ramos, 2006). *Data Mining* didefinisikan sebagai proses penemuan pola dalam data (Witten, 2011). *Data mining* sering juga disebut analisis data eksploratif. Data dalam jumlah besar yang diperoleh dari mesin kasir, pemindaian *barcode* dan dari berbagai basis data dalam perusahaan, kemudian ditelaah, dianalisis, dihapus dan dipakai ulang. Pencarian dilakukan pada model yang berbeda untuk memprediksi penjualan, respon pasar, keuntungan dan lain-lain (Olson & Shi, 2008).

Kemajuan dalam bidang *data mining* didorong oleh beberapa faktor, antara lain (Larose, 2005):

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam *data warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Sebuah standar lintas industri diperlukan dalam perusahaan untuk penggalian data yang dimiliki. *Cross-Industry Standard Process for Data Mining* (CRISP-DM) dikembangkan tahun 1996 oleh analis dari beberapa industri seperti DaimlerChrysler, SPSS dan NCR. CRISP-DM menyediakan standar proses *data*

*mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP-DM, Sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase (Gambar 2.1). Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah. Sebagai contoh, jika proses berada pada *fase modeling*, maka berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation* (Larose, 2005).



**Gambar 2.1** Proses Data Mining menurut CRISP-DM (Larose, 2005)

Enam fase CRSIP-DM (Larose, 2005):

1. Fase Pemahaman Bisnis (*Business / Research Understanding Phase*)
  - a. Penentuan tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
  - b. Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*.
  - c. Menyiapkan strategi awal untuk mencapai tujuan.
2. Fase Pemahaman Data (*Data Understanding Phase*)
  - a. Mengumpulkan data.
  - b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data

dan pencarian pengetahuan awal.

- c. Mengevaluasi kualitas data.
  - d. Jika diinginkan, pilih sebagian kecil group data yang mungkin mengandung pola dari permasalahan.
3. Fase Pengolahan Data (*Data Preparation Phase*)
    - a. Siapkan dari data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan berat yang perlu di laksanakan secara intensif.
    - b. Pilih kasus dan variabel yang ingin dianalisis dan yang sesuai analisis yang akan dilakukan.
    - c. Lakukan perubahan pada beberapa variable jika di butuhkan.
    - d. Siapkan data awal sehingga siap untuk perangkat permodelan.
  4. Fase Pemodelan (*Modeling Phase*)
    - a. Pilih dan aplikasikan teknik permodelan yang sesuai.
    - b. Kalibrasi aturan model untuk mengoptimalkan hasil.
    - c. Perlu diperhatikan bahwa beberapa teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.
    - d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.
  5. Fase Evaluasi (*Evaluation Phase*)
    - a. Mengevaluasi satu atau lebih model yang di gunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektifitas sebelum disebarkan untuk digunakan.
    - b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
    - c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
    - d. Mengambil keputusan berkaitan dengan penggunaan hasil dari *data mining*.
  6. Fase Penyebaran (*Deployment Phase*)
    - a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
    - b. Contoh sederhana penyebaran: Pembuatan laporan.
    - c. Contoh kompleks penyebaran: Penerapan proses *data mining* secara

paralel pada departemen lain.

*Data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Larose, 2005):

### 1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

### 2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

### 3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi menghasilkan nilai dari hasil di masa mendatang. Contoh prediksi dalam bisnis dan penelitian adalah:

- a. Prediksi harga beras dalam tiga bulan yang akan datang.

- b. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

#### 4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, pengolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah :

- a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
- b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
- c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

#### 5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidak miripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan atau homogen, yang mana kemiripan record dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan record dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah:

- a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.

- c. Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

#### 6. Asosiasi

Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:

- a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.
- b. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

#### 2.1.2 Klasifikasi

Klasifikasi adalah proses menempatkan obyek atau konsep tertentu kedalam satu set kategori, berdasarkan sifat obyek atau konsep yang bersangkutan (Gorunescu, 2011). Metode klasifikasi ditujukan untuk pembelajaran fungsi-fungsi berbeda yang memetakan masing-masing data terpilih kedalam salah satu dari kelompok kelas yang telah ditetapkan sebelumnya. Dari pembelajaran tersebut, metode klasifikasi dapat secara otomatis dapat memprediksi kelas dari data lain yang belum diklasifikasikan (Olson & Shi, 2008).

Proses klasifikasi didasarkan pada komponen (Gorunescu, 2011):

##### 1. Kelas (*class*)

Variabel dependen dari model yang merupakan kategori variabel yang mewakili label-label yang diletakkan pada obyek setelah pengklasifikasian. Contoh: kelas bintang dan kelas gempa bumi.

##### 2. Prediktor (*predictors*)

Variabel independen dari model yang diwakili oleh karakteristik atau atribut dari data yang diklasifikasikan berdasarkan klasifikasi yang dibuat. Contoh: tekanan darah, status perkawinan dan musim.

##### 3. Dataset pelatihan (*training dataset*)

Merupakan dataset yang berisi dua komponen nilai yang digunakan untuk pelatihan untuk mengenali model yang sesuai dengan kelasnya, berdasarkan

prediktor yang ada. Contoh: database penelitian gempa, database badai dan database pelanggan supermarket.

#### 4. Dataset pengujian (*testing dataset*)

Merupakan dataset baru yang akan diklasifikasikan oleh model yang dibangun sehingga dapat dievaluasi hasil akurasi klasifikasi tersebut.

### 2.1.3 Algoritma *Support Vector Machine*

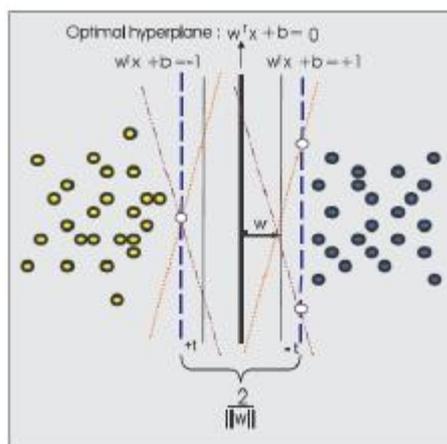
*Support Vector Machine (SVM)* diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992. SVM merupakan salah satu teknik yang relatif baru dibandingkan dengan teknik lain, tetapi memiliki performansi yang lebih baik di berbagai bidang aplikasi seperti bioinformatika, pengenalan tulisan tangan, klasifikasi teks, klasifikasi diagnosis penyakit dan lain sebagainya (Feng-Chia, 2009).

*Support Vector Machine (SVM)* adalah *metode learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space* (Bellotti & Crook, 2007). *Hyperplane* terbaik adalah *hyperplane* yang terletak ditengah-tengah antara dua set obyek dari dua *class*. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector* (Aydin, Karakose & Akin, 2011).

Yang menjadi karakteristik dari *Support Vector Machine (SVM)* adalah sebagai berikut:

1. Secara prinsip SVM adalah *linear classifier*.
2. *Pattern recognition* dilakukan dengan mentransformasikan data pada *input space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru tersebut. Hal ini membedakan SVM dari solusi *pattern recognition* pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi *input space*.
3. Menerapkan strategi *Structural Risk Minimization (SRM)*.
4. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua *class*.

Secara sederhana konsep SVM adalah sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*, dimana dapat dilihat pada gambar dibawah ini:



**Gambar 2.2 Konsep SVM untuk mencari *hyperplane* terbaik (Larose,2005)**

Pada gambar diatas memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class*: +1 dan -1. *Pattern* yang tergabung pada *class* -1 disimbolkan dengan warna kuning. Sedangkan *pattern* pada *class* +1, disimbolkan dengan warna biru. Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan garis berwarna orange. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur margin *hyperplane* tersebut, dan mencari titik maksimalnya.

Margin adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. *Hyperplane* yang terbaik yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik putih yang berada dalam garis bidang pembatas *class* adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.

Data yang tersedia dinotasikan sebagai  $x \in \mathbb{R}^d$  sedangkan label masing-masing dinotasikan  $y_i \in \{-1,+1\}$  untuk  $i = 1,2,\dots,l$  yang mana  $l$  adalah banyaknya

data. Diasumsikan kedua class  $-1$  dan  $+1$  dapat terpisah secara sempurna oleh hyperplane berdimensi  $d$ , yang didefinisikan:

$$w \cdot x + b = 0 \quad (2.1)$$

Sebuah pattern  $x_i$  yang termasuk class  $-1$  (sampel negatif) dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan:

$$w \cdot x + b = -1 \quad (2.2)$$

sedangkan pattern yang termasuk class  $+1$  (sampel positif):

$$w \cdot x + b = +1 \quad (2.3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya, yaitu  $1/\|w\|$ . Hal ini dapat dirumuskan sebagai Quadratic Programming (QP) problem, yaitu mencari titik minimal persamaan 2.4, dengan memperhatikan constraint persamaan 2.5.

$$\min_w \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.4)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0, \quad \forall_i \quad (2.5)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya Lagrange Multiplier sebagaimana ditunjukkan pada persamaan 2.6:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - w \sum_{i=1} \alpha_i (y_i (x_i \cdot w + b) - 1) \quad (i=1,2,\dots,l) \quad (2.6)$$

$\alpha_i$  adalah Lagrange multipliers, yang bernilai nol atau positif ( $\alpha_i \geq 0$ ). Nilai optimal dari persamaan (2.6) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$  dan  $b$ , dan memaksimalkan  $L$  terhadap  $\alpha_i$ . Dengan memperhatikan sifat bahwa pada titik optimal gradient  $L = 0$ , persamaan langkah 2.6 dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung saja  $\alpha_i$ , sebagaimana persamaan 2.7.

$$\text{Maximize: } \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i \cdot y_j \cdot x_i \cdot x_j \quad (2.7)$$

Subject to:

$$\alpha_i \geq 0 \quad (i=1,2,\dots,l) \quad \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.8)$$

Dari hasil dari perhitungan ini diperoleh  $\alpha_i$  yang kebanyakan bernilai positif. Data yang berkorelasi dengan  $\alpha_i$  yang positif inilah yang disebut sebagai *support vector*.

Sebagai contoh digunakan problem AND. Problem AND adalah klasifikasi dua kelas dengan empat data (lihat Tabel 2.1). Karena ini problem linier, kernelisasi tidak diperlukan.

**Table 2.1 AND Problem (Larose 2005)**

X1	X2	y
1	1	1
-1	1	-1
1	-1	-1
-1	-1	-1

dapatkan formulasi masalah optimisasi sebagai berikut:

$$\min \frac{1}{2} (w_1^2 + w_2^2) + C(t_1 + t_2 + t_3 + t_4)$$

Subject to

$$w_1 + w_2 + b + t_1 \geq 1$$

$$w_1 - w_2 - b + t_2 \geq 1$$

$$-w_1 + w_2 - b + t_3 \geq 1$$

$$w_1 + w_2 - b + t_4 \geq 1$$

$$t_1, t_2, t_3, t_4 \geq 0$$

Karena fungsi AND adalah kasus klasifikasi linier, maka bisa dipastikan nilai variable slack  $t_i = 0$ . Jadi Kita bisa masukkan nilai  $C = 0$ . Setelah menyelesaikan problem optimasi di atas didapat solusi

$$w_1 = 1, w_2 = 1, b = -1$$

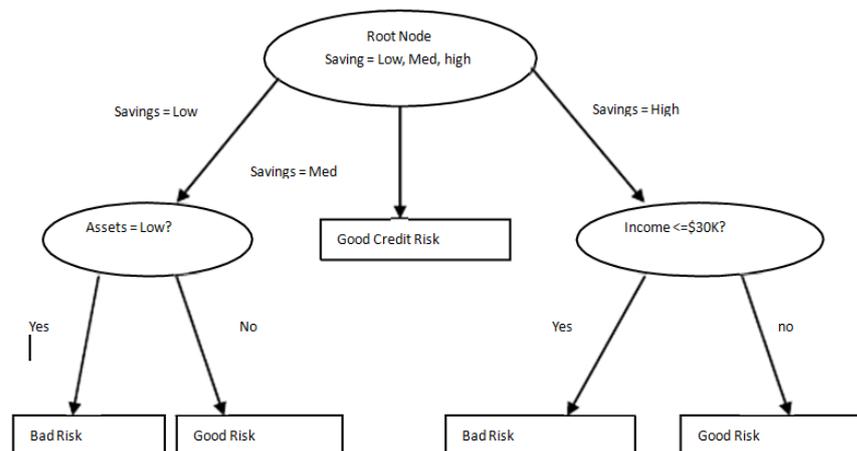
Persamaan fungsi pemisahnya adalah

$$f(x) = x_1 + x_2 - 1.$$

Untuk menentukan output atau label dari setiap titik data/obyek kita gunakan fungsi  $g(x) = \text{sign}(x)$ . Dengan fungsi sign ini semua nilai  $f(x) < 0$  diberi label  $-1$  dan lainnya diberi label  $+1$ .

#### 2.1.4 Algoritma C4.5

Salah satu metode klasifikasi menarik yang melibatkan konstruksi pohon keputusan, koleksi node keputusan, terhubung oleh cabang-cabang, memperpanjang bawah dari simpul akar samapai berakhir di node daun. Dimulai dari node root, yang oleh konvensi ditempatkan dibagian atas dari diagram pohon keputusan, atribut diuji pada node keputusan, dengan setiap hasil yang mungkin menghasilkan cabang. Setiap cabang kemudian mengarah ke node lain baik keputusan atau ke node daun untuk mengakhiri (Larose, 2005).



**Gambar 2.3 Contoh konsep pohon keputusan sederhana (Larose, 2005)**

Pada Gambar 2.3 variabel target untuk pohon keputusan adalah risiko kredit, dengan pelanggan potensial yang diklasifikasikan sebagai risiko kredit baik atau buruk. Variabel prediktor adalah saving (*low*, *med*, *high*), Assets (*low* atau non *low*), dan Income ( $\leq \$50.000$  atau  $> \$50.000$ ). Di sini, simpul akar merupakan simpul keputusan, pengujiannya apakah setiap memiliki tingkat tabungan (*saving*) yang rendah, sedang atau tinggi.

Algoritma C4.5 merupakan bagian dari kelompok algoritma *decision trees* dan merupakan katerogi 10 algoritma yang paling populer. Diakhir tahun 1970 hingga awal tahun 1980-an, J.Rosss Quinlan seorang peneliti dibidang mesin pembelajaran mengembangkan sebuah model keputusan yang dinamakan ID3 (*Iterative Dichotomiser*), walaupun sebelumnya proyek ini telah dibuat oleh E.B.

Hunt, J. Marin, dan P.T. Stone. Quinlan kemudian membuat algoritma C4.5 (pengembangan dari ID3) yang berbasis *supervised learning* (Han & Kamber, 2006).

Tahapan dalam membuat sebuah pohon keputusan dengan algoritma C4.5 (Larose, 2005) yaitu:

1. Mempersiapkan data training, data training biasanya diambil dari data histori yang pernah terjadi sebelumnya atau disebut data masa lalu dan sudah dikelompokkan dalam kelas-kelas tertentu.
2. Menghitung Total *Entropy* sebelum dicari masing-masing *Entropy class*

$$H(T) \tag{2.1}$$

$$\text{Keterangan: } = -\sum_j^j P_j \log_2(P_j)$$

H = Himpunan kasus

T = Atribut

P<sub>j</sub> = proporsi dari H<sub>j</sub> terhadap H

3. Hitung nilai Gain dengan information gain dengan rata-rata:

$$\text{Gain average } H(T) - H_{\text{saving}}(T) \tag{2.2}$$

=

Keterangan:

$H(T)$  = Total *Entropy*

$H_{\text{saving}}(T)$  = Total Gain information untuk masing-masing Atribut

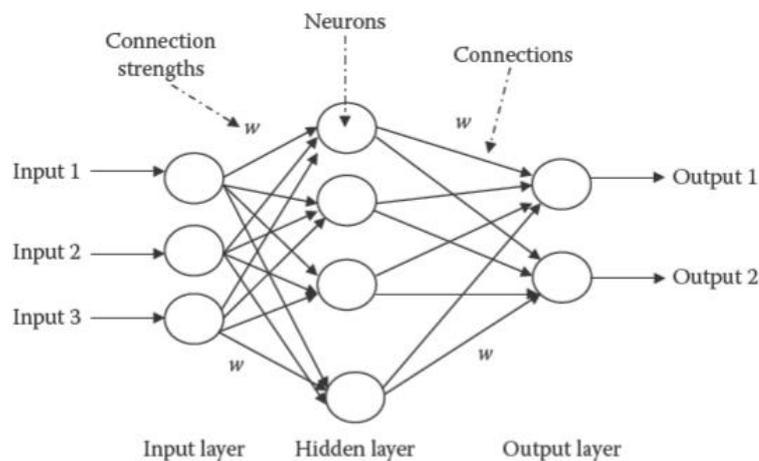
4. Ulangi langkah ke-2 dan ke-3 hingga semua tupel terpartisi

Proses partisi pohon keputusan akan berhenti disaat:

- a. Semua tupel dalam node N mendapatkan kelas yang sama
- b. Tidak ada atribut didalam tupel yang dipartisi lagi
- c. Tidak ada tupel didalam cabang yang kosong

### 2.1.5 Algoritma *Neural Network*

*Neural network* adalah suatu sistem pemroses informasi yang memiliki karakteristik menyerupai dengan jaringan saraf biologi pada manusia. *Neural network* didefinisikan sebagai sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut (Astuti, 2009). *Neural Network* (NN) atau Jaringan Syaraf Tiruan (JST) adalah suatu usaha untuk meniru fungsi otak manusia. Otak manusia diyakini terdiri dari jutaan unit pengolahan kecil, yang disebut *neuron*, yang bekerja secara paralel. *Neuron* saling terhubung satu sama lain melalui koneksi *neuron*. setiap individu *neuron* mengambil *input* dari satu set *neuron*. Ini kemudian memproses *input* tersebut dan melewati *output* untuk satu set *neuron*. Keluaran dikumpulkan oleh *neuron* lain untuk diproses lebih lanjut. Para otak manusia adalah jaringan kompleks *neuron* dimana koneksi tetap melanggar dan membentuk. Banyak model mirip dengan otak manusia telah diusulkan. (Shukla, Tiwari, & Kala, 2010)



Gambar 2.4 Arsitektur *Artificial Neural Network* (Shukla, 2010)

*Neural network* terdiri dari dua lapisan atau lebih, meskipun sebagian besar jaringan terdiri dari tiga lapisan: lapisan *input*, lapisan tersembunyi, dan lapisan *output* (Larose, 2005). Pendekatan *neural network* dimotivasi oleh jaringan saraf biologis. Secara kasar, *neural network* adalah satu set terhubung *input/output* unit, di mana masing-masing sambungan memiliki berat yang terkait dengannya. *Neural network* memiliki beberapa ciri yang membuat mereka populer untuk *clustering*. Pertama, *neural network* adalah arsitektur pengolahan *inheren paralel* dan terdistribusi. Kedua, *neural network* belajar dengan menyesuaikan bobot interkoneksi dengan data, Hal ini memungkinkan *neural network* untuk "menormalkan" pola dan bertindak sebagai fitur (atribut) *extractors* untuk kelompok yang berbeda. Ketiga, *neural network* memproses vektor numerik dan membutuhkan pola objek untuk diwakili oleh fitur kuantitatif saja (Gorunescu, 2011).

*Neural network* terdiri dari kumpulan node (neuron) dan relasi. Ada tiga tipe *node* (neuron) yaitu, *input*, *hidden* dan *output*. Setiap relasi menghubungkan dua buah node dengan bobot tertentu dan juga terdapat arah yang menunjukkan aliran data dalam proses (Kusrini & Luthfi, 2009).

Kemampuan otak manusia seperti mengingat, menghitung, mengeneralisasi, adaptasi, diharapkan *neural network* dapat meniru kemampuan otak manusia. *Neural network* berusaha meniru struktur/arsitektur dan cara kerja otak manusia sehingga diharapkan bisa dan mampu menggantikan beberapa pekerjaan manusia. *Neural network* berguna untuk memecahkan

persoalan yang berkaitan dengan pengenalan pola, klasifikasi, prediksi dan data mining (Shukla, Tiwari, & Kala, 2010).

Input *node* terdapat pada *layer* pertama dalam *neural network*. Secara umum setiap *input node* merepresentasikan sebuah input parameter seperti umur, jenis kelamin, atau jurusan. *Hidden node* merupakan *node* yang terdapat di bagian tengah. *Hidden node* ini menerima masukan dari *input node* pada *layer* pertama atau dari *hidden node* dari *layer* sebelumnya. *Hidden node* mengkombinasikan semua masukan berdasarkan bobot dari relasi yang terhubung, mengkalkulasikan, dan memberikan keluaran untuk *layer* berikutnya. *Output node* mempresentasikan atribut yang diprediksi (Kusrini & Luthfi, 2009).

Setiap *node* (neuron) dalam *neural network* merupakan sebuah unit pemrosesan. Tiap *node* memiliki beberapa masukan dan sebuah keluaran. Setiap *node* mengkombinasikan beberapa nilai masukan, melakukan kalkulasi, dan membangkitkan nilai keluaran (aktifasi). Dalam setiap *node* terdapat dua fungsi, yaitu fungsi untuk mengkombinasikan masukan dan fungsi aktifasi untuk menghitung keluaran. Terdapat beberapa metode untuk mengkombinasikan masukan antara lain *weighted sum*, *mean*, *max*, logika OR, atau logika AND (Kusrini & Luthfi, 2009). Serta beberapa fungsi aktifasi yang dapat digunakan yaitu *heaviside* (threshold), *step activation*, *piecewise*, *linear*, *gaussian*, *sigmoid*, *hyperbolic tangent* (Gorunescu, 2011).

Salah satu keuntungan menggunakan *neural network* adalah bahwa *neural network* cukup kuat sehubungan dengan data. Karena *neural network* berisi banyak *node* (*neuron* buatan) dengan bobot ditugaskan untuk setiap koneksi (Larose, 2005).

Menurut Astuti (2009) Algoritma *neural network* mempunyai karakteristik-karakteristik lainnya antara lain

1. Masukan dapat berupa nilai diskrit atau *real* yang memiliki banyak dimensi.
2. Keluaran berupa vektor yang terdiri dari beberapa nilai diskrit atau *real*.
3. Dapat mengetahui permasalahan secara *black box*, dengan hanya mengetahui nilai masukan serta keluarannya saja.
4. Mampu menangani pembelajaran terhadap data yang memiliki derau (*noise*)
5. Bentuk dari fungsi target pembelajaran tidak diketahui karena hanya berupa bobot-bobot nilai masukan pada setiap *neuron*.

6. Karena harus mengubah banyak nilai bobot pada proses pembelajaran, maka waktu pembelajaran menjadi lama, sehingga tidak cocok untuk masalah- masalah yang memerlukan waktu cepat dalam pembelajaran.
7. *Neural network* hasil pembelajaran tiruan dapat dijalankan dengan tepat.

Penelitian di bidang *neural network* dimulai pada masa komputer digital. McCulloch dan Pitts (1943) mengemukakan model matematika pertama untuk *neural network*. Rosenblatt (1962) mengemukakan model *perceptron* dan algoritma pembelajaran pada tahun 1962 (Alpaydin, 2010). Minsky dan Papert (1969) menunjukkan keterbatasan *single layer perceptron* untuk menyelesaikan masalah yang *nonlinearly separable*. Kemudian Rumelhart, Hinton, and Williams (1986) yang mempresentasikan algoritma *backpropagation* untuk *multilayer perceptron* yang dapat menyelesaikan masalah yang *nonlinearly separable* (Han & Kamber, 2007).

Aplikasi *neural network* telah banyak dimanfaatkan untuk berbagai kepentingan seperti di bidang Elektronik, Otomotif, Perbankan, Sistem penerbangan udara, dunia hiburan, transportasi publik, telekomunikasi, bidang kesehatan, keamanan, bidang robotika, asuransi, pabrik, *financial*, suara, pertambangan dan sistem pertahanan (Astuti, 2009)

Secara garis besar metode *learning* dibedakan menjadi dua, yaitu pendekatan *supervised* dan *unsupervised*. Dalam metode *unsupervised learning* metode diterapkan tanpa adanya latihan (*training*) dan tanpa guru (*teacher*), guru yang dimaksud disini adalah label dari data. Sedangkan dalam metode *supervised*, kita melatih dulu teknik yang digunakan dengan data yang ada, setelah dilatih dengan cukup data maka *tools* kita (misalkan *neuro network*) bisa digunakan untuk memprediksi keputusan apa yang harus diambil bila data baru masuk dan belum diketahui keluarannya.

Ada beberapa metode *supervised learning* yang telah ditemukan oleh para peneliti, diantaranya yang paling sering diaplikasikan adalah *perceptron* dan *backpropagation (feedforward)*.

## 1. Algoritma *Backpropagation*

Algoritma pelatihan *backpropagation* atau ada yang menterjemahkan menjadi propagasi balik pertama kali dirumuskan oleh Paul Werbos pada tahun 1974 dan dipopulerkan oleh Rumelhart bersama McClelland untuk dipakai pada *neural network*. Metode *backpropagation* pada awalnya dirancang untuk *neural network feedforward*, tetapi pada perkembangannya, metode ini diadaptasi untuk pembelajaran pada model *neural network* lainnya (Astuti, 2009). Salah satu metode pelatihan terawasi pada *neural network* adalah metode *backpropagation*, di mana ciri dari metode ini adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan.

Metode algoritma *backpropagation* ini banyak diaplikasikan secara luas. *backpropagation* telah berhasil diaplikasikan di berbagai bidang, antaranya bidang finansial, pengenalan pola tulisan tangan, pengenalan pola suara, sistem kendali, pengolah citra medika. *backpropagation* berhasil menjadi salah satu metode komputasi yang handal.

Algoritma *backpropagation* mempunyai pengaturan hubungan yang sangat sederhana yaitu: jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya galatnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan mendekati nilai yang benar. Algoritma ini juga berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*) (Purnomo & Kurniawan, 2006).

Inisialisasi awal bobot jaringan *backpropagation* yang terdiri atas lapisan *input*, lapisan tersembunyi, dan lapisan *output* (Astuti, 2009). Tahap pelatihan *backpropagation* merupakan langkah untuk melatih suatu *neural network* yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapis yang membentuk *neural network* melalui masing-masing unitnya). Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini disebut dengan fase *mapping* atau proses pengujian/*testing*.

Algoritma pelatihan *backpropagation* terdiri dari dua tahapan yaitu *feedforward* dan *backpropagation* dari galatnya. Langkah pembelajaran dalam algoritma *backpropagation* adalah sebagai berikut (Myatt, 2007):

- a. Langkah 1  
Inisialisasi bobot jaringan secara acak (biasanya antara -0.1 sampai 0.1)
- b. Langkah 2

Untuk setiap data pada data *training*, hitung *input* untuk simpul berdasarkan nilai *input* dan bobot jaringan saat itu, menggunakan rumus:

$$\text{Input } j = \sum_{i=1}^n O_i W_{ij} + \theta_j \quad (2.1)$$

c. Langkah 3

Berdasarkan *input* yang didapat dari langkah dua, selanjutnya membangkitkan *output* untuk simpul menggunakan fungsi aktivasi *sigmoid*:

$$\text{Output} = \frac{1}{1+e^{-\text{input}}} \quad (2.2)$$

d. Langkah 4

Hitung nilai *error* antara nilai yang diprediksi dengan nilai yang sesungguhnya menggunakan rumus :

$$\text{Error}_j = \text{output}_j \cdot (1 - \text{output}_j) \cdot (\text{Target}_j - \text{Output}_j) \quad (2.3)$$

e. Langkah 5

Setelah nilai *error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagation*) untuk menghitung nilai *error* pada *hidden layer*, menggunakan rumus:

$$\text{Error}_j = \text{Output}_j (1 - \text{Output}_j) \sum_{k=1}^n \text{Error}_k W_{jk} \quad (2.4)$$

f. Langkah 6

Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk memperbarui bobot relasi menggunakan rumus :

$$W_{ij} = W_{ij} + l \cdot \text{Error}_j \cdot \text{Output}_i \quad (2.5)$$

Keterangan Notasi:

$\text{Output}_j$  = Output aktual dari simpul j

$\text{Error}_k$  = error simpul k

$W_{jk}$  = Bobot relasi dari simpul j ke simpul k pada layer berikutnya

$\text{Output}_j$  = Output aktual dari simpul j

$\text{Target}_j$  = Nilai target yang sudah diketahui pada data training

$O_i$  = Output simpul  $i$  dari layer sebelumnya

$w_{ij}$  = bobot relasi dari simpul  $i$  pada layer sebelumnya ke simpul  $j$

$\theta_j$  = bias (sebagai pembatas)

$W_{ij}$  = bobot relasi dari unit  $i$  pada layer sebelumnya ke unit  $j$

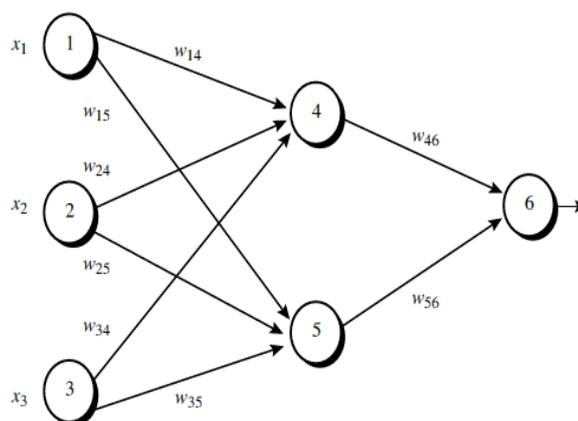
$l$  = learning rate (konstanta, nilainya antara 0 sampai dengan 1)

Error $_j$  = Error pada output layer simpul  $j$

Output $_i$  = Output dari simpul  $i$

## 2. Contoh Penerapan *Neural Network*

Berikut adalah contoh kasus penyelesaian masalah menggunakan *neural network* dengan metode *backpropogation* (Han & Kamber, 2007)



Gambar 2.5 Contoh Arsitektur

*Neural Network Multilayer feed-forward*(Han & Kamber,2007)

### 1. Tahap 1.

Inisialisasikan nilai awal dan nilai bias. Misalkan nilai *learning rate* ( $l$ ) adalah 0.9 dan inialisasi nilai awal dan nilai bias, pada training data pertama  $X=(1,0,1)$ , yang kelas labelnya adalah 1.

Tabel 2.2 Input Nilai bias, bobot dan bias (Han & Kamber,2007)

$X_1$	$X_2$	$X_3$	$W_{21}$	$W_{22}$	$W_{31}$	$W_{32}$	$W_{41}$	$W_{42}$	$W_{51}$	$W_{61}$	$b_1$	$b_2$	$b_3$
1	0	1	0.2	-0.3	0.1	0.4	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

### 2. Tahap 2.

Hitung nilai *input*, *output* dan *error* untuk setiap node.

Rumus perhitungan input:

$$\text{Input } 2 = (X_1 \cdot W_{21}) + (X_2 \cdot W_{31}) + (X_3 \cdot W_{41}) + b_1$$

$$\begin{aligned}
&= (1 \cdot 0.2) + (0 \cdot 0.4) + (1 \cdot -0.5) + -0.4 \\
&= 0.2 + 0 + (-0.5) + (-0.4) \\
&= -0.7
\end{aligned}$$

### 3. Tahap 3

Membangkitkan *output* untuk simpul menggunakan fungsi aktivasi sigmoid

$$Output = \frac{1}{1 + e^{-Input}}$$

$$\begin{aligned}
Output\ 2 &= 1/(1 + e^{0.7}) \\
&= 1/(1 + 2.718280.7) \\
&= 0.332
\end{aligned}$$

$$\begin{aligned}
Input\ 3 &= (X_1 \cdot W_{22}) + (X_2 \cdot W_{32}) + (X_3 \cdot W_{42}) + b_2 \\
&= (1 \cdot (-0.3)) + (0 \cdot 0.1) + (1 \cdot 0.2) + 0.2 \\
&= (-0.3) + 0 + 0.2 + 0.2 \\
&= 0.1
\end{aligned}$$

$$\begin{aligned}
Output\ 3 &= 1/(1 + e^{0.1}) \\
&= 1/(1 + 2.718280.1) \\
&= 0.525
\end{aligned}$$

$$\begin{aligned}
Input\ 4 &= (output\ 2 \cdot W_{51}) + (output\ 3 \cdot W_{61}) + b_3 \\
&= (0.332 \cdot -0.3) + (0.525 \cdot -0.2) + 0.1 \\
&= -0.105
\end{aligned}$$

$$\begin{aligned}
Output\ 4 &= 1/(1 + e^{-0.105}) \\
&= 1/(1 + 2.718280^{-0.105})
\end{aligned}$$

4. Hitung nilai *error* antara nilai yang diprediksi dengan nilai sesungguhnya. Dan langkah ke -5 dibalik kelayar sebelumnya untuk menghitung nilai error pada hidden layer

Rumus perhitungan error:

$$Error_j = Output_j(1 - Output_j) \sum_{k=1}^n Error_k w_{jk}$$

$$\begin{aligned}
Error\ 4 &= output4 \cdot (1 - output4) \cdot (outputtarget - output4) \\
&= 0.474 \cdot (1 - 0.474) \cdot (1 - 0.474) \\
&= 0.474 \cdot 0.526 \cdot 0.526
\end{aligned}$$

$$= 0.1311$$

$$\text{Error}_3 = \text{output}_3 \cdot (1 - \text{output}_3) \cdot \text{error}_4 \cdot W_{61}$$

$$= 0.525 \cdot (1 - 0.525) \cdot 0.1311 \cdot -0.2$$

$$= 0.525 \cdot 0.475 \cdot 0.1311 \cdot -0.2$$

$$= -0.0065$$

$$\text{Error}_2 = \text{output}_2 \cdot (1 - \text{output}_2) \cdot \text{error}_4 \cdot W_{51}$$

$$= 0.332 \cdot (1 - 0.332) \cdot 0.1311 \cdot -0.3$$

$$= 0.332 \cdot 0.668 \cdot 0.1311 \cdot -0.3$$

$$= -0.0087$$

6. Hitung nilai bobot dan nilai bias terbaru untuk setiap relasi antara *hidden layer*

dan *output layer*. Untuk perhitungan nilai bobot dan bias baru menggunakan *learning rate* 0.9.

Tabel 2.2 adalah tabel nilai bias dan bobot terbaru, perhitungan nilai bias dan bobot terbaru dengan menggunakan rumus:

$$w_{ij} = w_{ij} + l \cdot \text{Error}_j \cdot \text{Output}_i$$

Tabel 2.3 Perhitungan untuk bias dan bobot baru (Han & Kamber,2007)

Bobot atau bias	Nilai Baru
$W_{51}$	$-0.3+(0.9)(0.1311)(0.332) = -0.261$
$W_{61}$	$-0.2+(0.9)(0.525)(-0.138) = -0.138$
$W_{21}$	$0.2+(0.9)(-0.0087)(1) = 0.912$
$W_{22}$	$-0.3+(0.9)(-0.0065)(1) = -0.306$
$W_{31}$	$0.4+(0.9)(-0.0087)(0) = 0.4$
$W_{32}$	$0.1+(0.9)(-0.0065)(0) = 0.1$
$W_{41}$	$-0.5+(0.9)(-0.0087)(1) = -0.508$
$W_{42}$	$0.2+(0.9)(-0.0065)(1) = 0.194$
$b_1$	$0.1+(0.9)(0.1311) = 0.218$
$b_2$	$0.2+(0.9)(-0.0065) = 0.194$
$b_3$	$-0.4+(0.9)(-0.0087) = -0.408$

### 2.1.6 Seleksi Atribut

Proses *data mining* membutuhkan biaya komputasi yang tinggi ketika berhadapan dengan kumpulan data dalam jumlah besar. Mengurangi dimensi yaitu jumlah atribut set data atau kelompok atribut, secara efektif dapat memotong

biaya tersebut. Pengurangan dimensi tersebut dilakukan dengan menekan seminimal mungkin kerugian yang dapat terjadi akibat kehilangan sebagian informasi. Tujuan pengurangan dimensi dalam domain *data mining* adalah untuk mengidentifikasi biaya terkecil di mana algoritma *data mining* dapat menjaga tingkat kesalahan di bawah perbatasan garis efisiensi. Yang dimaksud dengan biaya adalah fungsi dari kompleksitas teoritis dari algoritma *data mining* yang berasal dari model, dan berkorelasi dengan waktu yang dibutuhkan algoritma tersebut dalam menjalankan model, serta ukuran dari kumpulan data (Maimon & Rokach, 2010).

Seleksi atribut adalah masalah terkait erat dengan pengurangan dimensi. Tujuan seleksi atribut adalah untuk mengidentifikasi tingkat kepentingan atribut dalam kumpulan data, dan membuang semua atribut lain seperti informasi yang tidak relevan dan berlebihan. Karena seleksi atribut mengurangi dimensi dari data, maka hal ini akan memungkinkan operasi algoritma data mining dapat berjalan lebih efektif dan lebih cepat. Dalam beberapa kasus dengan dilakukannya seleksi atribut dihasilkan peningkatan tingkat akurasi klasifikasi (Maimon & Rokach, 2010).

Di sisi lain, seleksi atribut adalah proses yang mahal, dan juga bertentangan dengan asumsi awal yaitu bahwa semua informasi atau atribut diperlukan dalam rangka mencapai akurasi maksimal.

Ada empat alasan utama untuk melakukan pengurangan dimensi yaitu (Maimon & Rokach, 2010):

1. Penurunan biaya model pembelajaran
2. Meningkatkan kinerja model pembelajaran
3. Mengurangi dimensi yang tidak relevan
4. Mengurangi dimensi yang berlebihan

Tujuan seleksi atribut adalah untuk pengurangan atribut dari dataset untuk menghilangkan variabel yang dianggap tidak relevan. Metode seleksi fitur dapat diklasifikasikan ke dalam tiga kategori utama (Vercellis, 2009):

1. Metode *filter*

Metode *Filter* adalah memilih atribut yang relevan sebelum pindah ke tahap pembelajaran berikutnya, atribut yang dianggap paling penting yang dipilih untuk pembelajar, sedangkan sisanya dikecualikan.

2. Metode *wrapper*

Metode *wrapper* menilai sekelompok variabel dengan menggunakan klasifikasi yang sama atau algoritma regresi digunakan untuk memprediksi nilai dari variabel target.

### 3. Metode *embedded*

Untuk metode *embedded*, proses seleksi atribut terletak di dalam algoritma pembelajaran, sehingga pemilihan set optimal atribut secara langsung dibuat selama fase generasi model.

#### 2.1.7 Algoritma *Particle Swarm Optimization*

Algoritma PSO pertama kali diusulkan oleh Kennedy dan Eberhart pada tahun 1995. *Particle swarm optimization (PSO)* adalah jenis algoritma kecerdasan yang berasal dari perilaku kawanan burung mencari makan (Yun, Qiu-yan & Hua, 2011). *Particle swarm optimization (PSO)* merupakan teknik komputasi evolusioner yang mampu menghasilkan solusi secara global optimal dalam ruang pencarian melalui interaksi individu dalam segerombolan partikel. Setiap partikel menyampaikan informasi berupa posisi terbaiknya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang terbaik tersebut (Shuzhou & Bo, 2011).

*Particle swarm optimization (PSO)* dapat diasumsikan dengan sekelompok burung yang secara acak mencari makanan di suatu daerah. Hanya ada satu potong makanan di daerah yang dicari tersebut. Burung-burung tidak tahu di mana makanan tersebut. Tapi mereka tahu seberapa jauh makanan tersebut dan posisi rekan-rekan mereka. Jadi strategi terbaik untuk menemukan makanan adalah dengan mengikuti burung yang terdekat dari makanan (Abraham, Grosan & Ramos, 2006).

Untuk menemukan solusi yang optimal, maka setiap partikel akan bergerak ke arah posisi yang terbaik sebelumnya (*pbest*) dan posisi terbaik secara global (*gbest*). Sebagai contoh, partikel ke-*i* dinyatakan sebagai :  $x_i = (x_{i,1}, x_{i,2} \dots x_{i,d})$  dalam ruang *d*-dimensi. Posisi terbaik sebelumnya dari partikel ke-*i* disimpan dan dinyatakan sebagai  $pbest_i = (pbest_{i,1}, pbest_{i,2}, \dots pbest_{i,d})$ . Modifikasi kecepatan dan posisi tiap partikel dapat dihitung menggunakan kecepatan saat ini dan jarak  $pbest_{i,d}$  ke  $pbest_d$  seperti ditunjukkan oleh persamaan berikut:

$$v_{i,m} = w.v_{i,m} + c_1 * R * (pbest_{i,m} - x_{i,m}) + c_2 * R * (gbest_m - x_{i,m}) \quad (2.9)$$

$$x_{id} = x_{i,m} + v_{i,m} \quad (2.10)$$

Dimana:

$n$  : jumlah partikel dalam kelompok

$d$  : dimensi

$v_{i,m}$  : kecepatan partikel ke- $i$  pada iterasi ke- $i$

$w$  : faktor bobot inersia

$c_1, c_2$  : konstanta akselerasi (*learning rate*)

$R$  : bilangan random (0-1)

$x_{i,d}$  : posisi saat ini dari partikel ke- $i$  pada iterasi ke- $i$

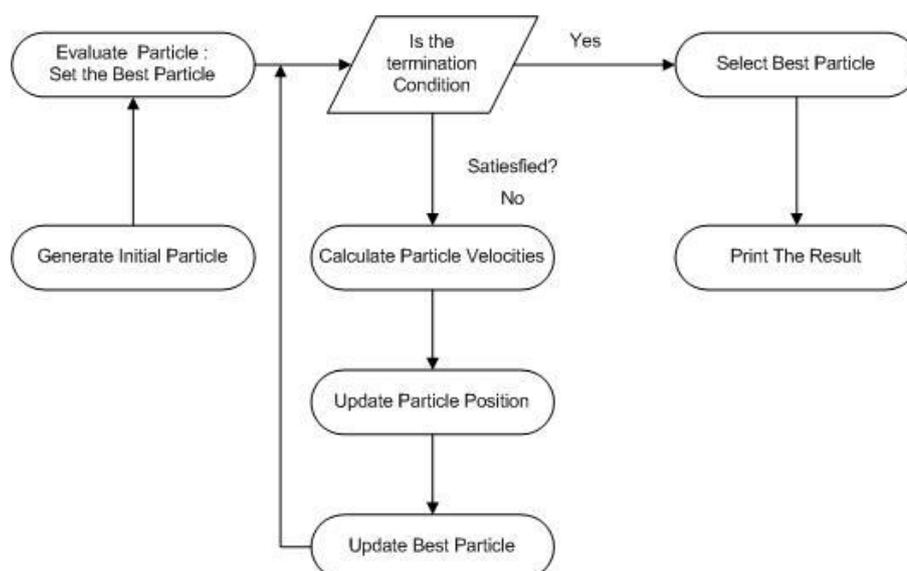
$pbest_i$  : posisi terbaik sebelumnya dari partikel ke- $i$

$gbest$  : partikel terbaik diantara semua partikel dalam satu kelompok atau populasi

Persamaan (2.9) menghitung kecepatan baru untuk tiap partikel (solusi potensial) berdasarkan pada kecepatan sebelumnya ( $v_{i,m}$ ), lokasi partikel dimana nilai *fitness* terbaik telah dicapai ( $pbest$ ), dan lokasi populasi global ( $gbest$  untuk versi global,  $lbest$  untuk versi local) atau *local neighborhood* pada algoritma versi local dimana nilai *fitness* terbaik telah dicapai.

Persamaan (2.10) memperbaharui posisi tiap partikel pada ruang solusi. Dua bilangan acak  $c_1$  dan  $c_2$  dibangkitkan sendiri. Penggunaan berat inersia  $w$  telah memberikan performa yang meningkat pada sejumlah aplikasi.

Secara garis besar, struktur dasar dari PSO dapat digambarkan dalam bagan dibawah ini:



**Gambar 2.6 Struktur dasar PSO (Abraham, Grosan & Ramos, 2006)**

### 2.1.8 Pengujian *K-Fold Cross Validation*

*Cross Validation* adalah teknik validasi dengan membagi data secara acak kedalam k bagian dan masing-masing bagian akan dilakukan proses klasifikasi (Han & Kamber, 2006). Dengan menggunakan *cross validation* akan dilakukan percobaan sebanyak k. Data yang digunakan dalam percobaan ini adalah data training untuk mencari nilai error rate secara keseluruhan. Secara umum pengujian nilai k dilakukan sebanyak 10 kali untuk memperkirakan akurasi estimasi. Dalam penelitian ini nilai k yang digunakan berjumlah 10 atau *10-fold Cross Validation*. tiap percobaan akan menggunakan satu *data testing* dan k-1 bagian akan menjadi *data training*, kemudian *data testing* itu akan ditukar dengan satu buah *data training* sehingga untuk tiap percobaan akan didapatkan *data testing* yang berbeda-beda.

### 2.1.9 Evaluasi dan Validasi Hasil

#### 2.1.9.1 *Confusion matrix*

*Confusion matrix* memberikan keputusan yang diperoleh dalam *training* dan *testing*, *confusion matrix* memberikan penilaian *performance* klasifikasi berdasarkan objek dengan benar atau salah (Gorunescu, 2011). *Confusion matrix* berisi informasi aktual (*actual*) dan prediksi (*predicted*) pada sistem klasifikasi.

Tabel 2.4 Confusion Matrix(Gorunescu, 2011)

Classification	Predicted Class		
		Class = Yes	Class = No
Observed Class	Class = Yes	<i>a</i> <i>true positive –</i> TP)	<i>b</i> <i>(false negative –</i> FN)
	Class = No	<i>c</i> <i>(false positive –</i> FP)	<i>d</i> <i>(true negative –</i> TN)

Keterangan:

*True Positive* (TP) = proporsi positif dalam data set yang diklasifikasikan positif

*True Negative* (TN) = proporsi negative dalam data set yang diklasifikasikan negative

*False Positive* (FP) = proporsi negatif dalam data set yang diklasifikasikan positif

*False Negative* (FN) = proporsi negative dalam data set yang diklasifikasikan negatif

Berikut adalah persamaan model *confusion matrix*:

- a. Nilai *Accuracy* adalah proporsi jumlah prediksi yang benar. Dapat dihitung dengan menggunakan persamaan:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- b. *Sensitivity* digunakan untuk membandingkan proporsi TP terhadap tupel yang positif, yang dihitung dengan menggunakan persamaan:

$$Sensitivity = \frac{TP}{TP + FN}$$

- c. *Specificity* digunakan untuk membandingkan proporsi TN terhadap tupel yang negatif, yang dihitung dengan menggunakan persamaan:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- d. PPV (*positive predictive value*) adalah proporsi kasus dengan hasil diagnosa positif, yang dihitung dengan menggunakan persamaan:

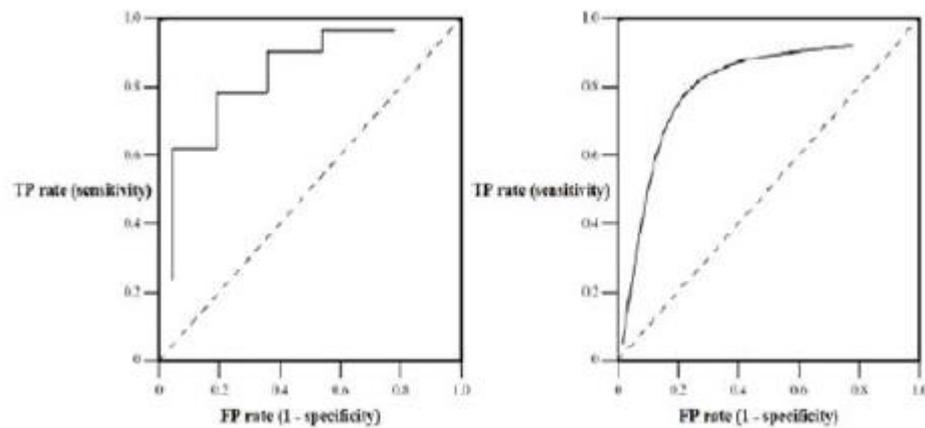
$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- e. NPV (*negative predictive value*) adalah proporsi kasus dengan hasil diagnosa negatif, yang dihitung dengan menggunakan persamaan:

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}}$$

### 2.1.9.2 Kurva ROC

Kurva ROC (*Receiver Operating Characteristic*) adalah alat visual yang berguna untuk membandingkan dua model klasifikasi. ROC mengekspresikan *confusion matrix*. ROC adalah grafik dua dimensi dengan *false positives* sebagai garis horisontal dan *true positives* sebagai garis vertikal (Vecellis, 2009). Dengan kurva ROC, kita dapat melihat *trade off* antara tingkat dimana suatu model dapat mengenali tuple positif secara akurat dan tingkat dimana model tersebut salah mengenali tuple negatif sebagai tuple positif. Sebuah grafik ROC adalah plot dua dimensi dengan proporsi positif salah (fp) pada sumbu X dan proporsi positif benar (tp) pada sumbu Y. Titik (0,1) merupakan klasifikasi yang sempurna terhadap semua kasus positif dan kasus negatif. Nilai positif salah adalah tidak ada (fp = 0) dan nilai positif benar adalah tinggi (tp = 1). Titik (0,0) adalah klasifikasi yang memprediksi setiap kasus menjadi negatif {-1}, dan titik (1,1) adalah klasifikasi yang memprediksi setiap kasus menjadi positif {1}. Grafik ROC menggambarkan *trade-off* antara manfaat (*true positive*) dan biaya (*false positives*). Berikut tampilan dua jenis kurva ROC (*discrete* dan *continuous*).



**Gambar 2.7** Grafik ROC (*discrete* dan *continous*) (Gorunescu, 2011)

Poin diatas garis diagonal merupakan hasil klasifikasi yang baik, sedangkan point

dibawah garis diagonal merupakan hasil klasifikasi yang buruk.

Dapat

disimpulkan bahwa, satu point pada kurva ROC adalah lebih baik dari pada yang

lainnya jika arah garis melintang dari kiri bawah ke kanan atas didalam grafik.

Untuk tingkat akuransi nilai AUC dalam klasifikasi *data mining* dibagi menjadi

lima kelompok (Gorunescu, 2011), yaitu:

- a. 0.90 - 1.00 = klasifikasi sangat baik (*excellent classification*)
- b. 0.80 - 0.90 = klasifikasi baik (*good classification*)
- c. 0.70 - 0.80 = klasifikasi cukup (*fair classification*)
- d. 0.60 - 0.70 = klasifikasi buruk (*poor classification*)
- e. 0.50 - 0.60 = klasifikasi salah (*failure*)

### 2.1.10 Standar ISO 9126

Kualitas perangkat lunak dapat dinilai melalui ukuran-ukuran dan metode-metode tertentu, serta melalui pengujian-pengujian software. Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software (Jung Ho-Wong, Kim Seung-Gweon, 2004). Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
6. *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

### 2.1.11 Penilaian Kredit

Menurut UU Perbankan No.10 Tahun 1998 Pasal 1 Ayat 11 menyatakan bahwa kredit adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga (UU Perbankan No.10 Tahun 1998).

Penilaian kredit didefinisikan sebagai teknik dan model keputusan yang mendasari penyedia kredit memutuskan masalah pemberian kredit kepada konsumen (Heiat, 2011). Penilaian kredit ini merupakan kumpulan data nasabah yang diambil dari data aplikasi pinjaman nasabah. Penilaian kredit mengelompokkan para calon debitur menjadi dua jenis yaitu debitur baik dan debitur buruk. Debitur baik memiliki kemungkinan besar akan membayar kewajiban keuangannya dengan lancar, sedangkan debitur buruk memiliki kemungkinan besar mengalami kredit macet (Gang, Jinxing, Jian & Hongbing, 2011). Di Indonesia saat ini, data *credit report* debitur dapat dilihat dari data yang dimiliki oleh Bank Indonesia yang dikelola oleh Biro Informasi Kredit. Penyedia kredit dapat memanfaatkan *credit report* ini untuk menganalisa permohonan kredit nasabah.

Salah satu resiko kredit yang harus dihindari adalah kredit macet. Hal ini terjadi apabila dalam jangka panjang penyedia kredit tidak mampu menarik pinjaman kredit dalam waktu yang telah ditentukan. Apabila hal ini dibiarkan berlarut-larut, maka dapat mengakibatkan penghentian operasional bahkan mengarah kepada kebangkrutan perusahaan penyedia kredit (Jianguo & Tao, 2008). Oleh karena itu, sangat penting dilakukan penilaian kredit terhadap permohonan kredit nasabah.

Tujuan dari penilaian kredit adalah untuk membantu penyedia kredit mengukur dan mengelola risiko keuangan dalam memberikan kredit sehingga mereka dapat membuat keputusan yang lebih baik, lebih cepat dan lebih obyektif (Hian, Wei & Chwee, 2006). Manfaat dari penilaian kredit antara lain dapat membantu mengurangi diskriminasi karena model penilaian kredit memberikan analisis obyektif dari kelayakan kredit seorang konsumen, untuk meningkatkan

kecepatan dan konsistensi dari proses aplikasi pinjaman dan memungkinkan otomatisasi proses kredit, dapat membantu lembaga keuangan dalam membuat keputusan yang berkualitas, lebih cepat, lebih baik, dan lebih tinggi akurasi, menetapkan tingkat bunga dan batas kredit yang akan ditetapkan untuk konsumen (Hian, Wei & Chwee, 2006).

Penilaian kredit dapat diaplikasikan dalam berbagai bidang seperti dalam industri asuransi menggunakan penilaian kredit untuk menyederhanakan aplikasi dan proses perpanjangan, menentukan premi dan prediksi akuntabilitas konsumen yang lebih akurat. Lembaga keuangan menggunakan penilaian kredit untuk memutuskan kelayakan pemberian kredit. Tuan tanah dapat menggunakan penilaian kredit untuk memprediksi apakah penyewa akan membayar tepat waktu atau tidak. *Supplier* juga menggunakan penilaian kredit untuk menetapkan konsumen potensial (Hian, Wei & Chwee, 2006).

Dalam prakteknya, terdapat dua kesalahan yang sering terjadi dalam penilaian kredit. Kesalahan tipe satu yaitu konsumen baik diidentifikasi sebagai konsumen buruk sehingga permohonan kredit ditolak, yang kedua yaitu kesalahan tipe dua dimana konsumen buruk diidentifikasi sebagai konsumen yang baik sehingga permohonan kredit diterima, kesalahan tipe dua inilah yang menyebabkan penyedia kredit mengalami kerugian (Ming-hui & Xu-chuang, 2007). Analisis kredit dilakukan oleh *account officer* yang dari sisi level jabatannya merupakan level seksi atau bagian, atau bahkan dapat berupa tim (*committee*) yang ditugaskan untuk melakukan analisis permohonan kredit (Edward, 2006). Dimana prinsip dasar dalam menganalisis kredit yang lazim dikenal dengan Prinsip 6 C's, yaitu (Edward, 2006):

#### 1. *Character*

*Character* adalah keadaan watak atau sifat dari nasabah, baik dalam kehidupan pribadi maupun dalam lingkungan usaha. Kegunaan dari penilaian terhadap karakter ini adalah untuk mengetahui sampai sejauh mana itikad atau kemauan nasabah untuk memenuhi kewajibannya (*willingness to pay*) sesuai dengan perjanjian yang telah ditetapkan.

### 2. *Capital*

*Capital* adalah jumlah dana atau modal sendiri yang dimiliki oleh calon nasabah. Semakin besar modal sendiri dalam perusahaan, tentu semakin tinggi kesungguhan calon nasabah dalam menjalankan usahanya dan lembaga pemberi kredit akan merasa lebih yakin dalam memberikan kredit.

### 3. *Capacity*

*Capacity* adalah kemampuan yang dimiliki calon nasabah dalam menjalankan usahanya guna memperoleh laba yang diharapkan. Kegunaan dari penilaian ini adalah untuk mengetahui atau mengukur sampai sejauh mana calon nasabah mampu untuk mengembalikan atau melunasi utang-utang (*ability to pay*) secara tepat dari usahanya yang diperolehnya.

### 4. *Collateral*

*Collateral* adalah barang-barang yang diserahkan nasabah sebagai agunan terhadap kredit yang diterimanya. *Collateral* tersebut harus dinilai oleh bank untuk mengetahui sejauh mana dari usaha yang diperolehnya.

### 5. *Condition of Economic*

*Condition of Economic* yaitu situasi dan kondisi politik, sosial, ekonomi, budaya yang memengaruhi keadaan perekonomian pada suatu saat yang kemungkinannya memengaruhi kelancaran perusahaan calon kreditur.

### 6. *Constrain*

*Constrain* adalah batasan dan hambatan yang tidak memungkinkan suatu bisnis untuk dilaksanakan pada tempat tertentu, misalkan pendirian suatu usaha pompa bensin yang disekitarnya banyak bengkel las atau pembakaran batu baru.

## 2.2 Tinjauan Studi

Beberapa penelitian terdahulu yang membahas tentang penentuan kelayakan pemberian kredit kepada nasabah diantaranya :

Penelitian yang dilakukan oleh Yi Jiang, et al. yang berjudul *A Bank Customer Credit Evaluation Based on the Decision Tree and the Simulated Annealing Algorithm* yang bertujuan untuk penentuan nasabah baik dan buruk dalam pemberian kredit, dalam penelitian ini metode yang dilakukan adalah

melakukan training terhadap 1000 data dengan metode *Decision Tree* dan metode *Simulated Annealing Algorithm*(Yi Jian, et al. 2007)

Penelitian yang dilakukan oleh Ning Liu, En Jun Xia, Li yang berjudul *Research and Application of PSO-BP Neural Network in Credit Risk Assessment*, dalam penelitian ini menggunakan metode perbandingan antara *Genetic Algorithm BP Neural Network* dengan *PSO BP Neural Network* untuk mendapatkan hasil yang akurat mengenai penentuan kelayakan pemberian kredit. hasil dari penelitian ini adalah PSO + BPNN lebih baik dikarenakan memiliki tingkat eror lebih sedikit yaitu 1,6487% dibandingkan GA + BPNN dengan tingkat eror 3,3163%(Ning Liu, En Jun Xia, Li Yang, 2010).

Penelitian yang dilakukan Yun Li, Qiu Yancoo, Hua Zhang yang berjudul *Application of the PSO - SVM Model for Credit Scoring*, dalam penelitian ini menggunakan metode perbandingan beberapa algoritma diantaranya GA +SVM, GS + SVM, PSO + SVM, DT + SVM, LDA + SVM, RS + SVM dengan menggunakan Dataset dari *UCI Repository* Jerman dan Australia. hasil dari penelitian ini PSO +SVM memiliki akurasi yang lebih baik dibandingkan metode lainnya yaitu untuk dataset Jerman 78,70% dan untuk dataset Australia 87,10 % (Yun Li, Qiu Yancoo, Hua Zhang, 2011).

Tabel 2.5 Penelitian Sebelumnya yang Terkait

<b>Keterangan</b>	C4.5	PSO + BP NN	PSO + SVM
<b>Jumlah Atribut</b>	20 Atribut	11 Atribut	Data UCI
<b>Permasalahan</b>	Penentuan nasabah baik dan buruk dalam hal analisis pemberian pinjaman	Peningkatan akurasi dan keandalan dalam penentuan resiko pemberian kredit	Mengurangi resiko dalam prediksi pemberian kredit kepada nasabah
<b>Peneliti</b>	Yi Jian, Yan Chen, Zhimin Zheng dan Xiang Jian He	Ning Liu, En Jun Xia, Li Yang	Yun Li, Qiu Yancoo, Hua Zhang
<b>Objek Penelitian</b>	<i>Credit Risk</i>	<i>Credit Risk</i>	<i>Credit Risk</i>

<b>Metode Penelitian</b>	Meningkatkan akurasi pengklasifikasian nasabah baik dan nasabah buruk dengan menggunakan C4.5 dan SAA	Membandingkan algoritma GA + BP NN dengan PSO + BP NN	Membandingkan beberapa algoritma diantaranya GA +SVM, GS + SVM, PSO + SVM, DT + SVM, LDA + SVM, RS + SVM
<b>Hasil Penelitian</b>	Algoritma c4.5 memiliki tingkat akurasi yang lebih tinggi yaitu hanya memiliki eror sekitar 24 % sedangkan SAA memiliki eror sebanyak 34 %	PSO + BPNN lebih baik dari GA +BPNN dikarenakan memiliki tingkat eror lebih sedikit yaitu 1,6487% serta memiliki kecepatan dalam memproses.	PSO +SVM memiliki akurasi yang lebih baik dibandingkan metode lainnya yaitu untuk dataset Jerman 78,70% dan untuk dataset Australia 87,10 %

Dari tabel di atas dapat disimpulkan bahwa ada beberapa metode yang dapat dilakukan untuk menentukan kelayakan pemberian kredit. dari metode metode yang ada di atas dalam penelitian kali ini, peneliti ingin melakukan komparasi terhadap ke tiga metode yang pada penelitian terkait diatas merupakan algoritma terakurat yaitu algoritma C4.5, SVM, dan *Neural Network* dengan menambahkan *Partical Swarm Optimazation* sebagai fitur seleksi atribut untuk mendapatkan hasil terbaik dalam penentuan kelayakan pemberian kredit. Penelitian yang saya lakukan berbeda dengan penelitian sebelumnya dikarenakan pada penelitian yang dilakukan oleh :

1. Penelitian Yi Jian et al adalah mengkomparasi 2 algoritma yaitu C4.5 dan SAA.
2. penelitian yang dilakukan Ning Liu, En Jun Xia, Li Yang merupakan penelitian yang mengoptimasi algoritma *Neural Network* dengan algoritma *Particle Swarm Optimazation* dan *Genetic Algorithm*.
3. penelitian yang dilakukan oleh Yun Li, Qiu Yancoo, Hua Zhang merupakan penelitian yang mengoptimasi algoritma *Support Vector Machine* dengan GA, PSO, DT, LDA, RS, dan GS.

## **2.3 Tinjauan Perusahaan**

### **2.3.1 Sejarah Bank DKI Jakarta**

Bank DKI merupakan Bank Umum dan Badan Usaha Milik Daerah yang kepemilikan sahamnya dimiliki oleh Pemerintah Provinsi DKI Jakarta dan PD Pasar Jaya. Bank DKI kini telah tumbuh menjadi Bank Umum beraset Rp 30,74 triliun, dengan jaringan kantor mencapai 221 kantor yang tersebar di Jakarta dan beberapa kota besar di Indonesia, dengan sumber daya manusia 2.549 karyawan serta ragam produk dan layanan perbankan kepada lebih dari 1 juta nasabah.

Bank DKI pertama kali didirikan di Jakarta dengan nama “PT Bank Pembangunan Daerah Djakarta Raya” sebagaimana termaktub dalam Akta Pendirian Perseroan Terbatas Perusahaan Bank Pembangunan Daerah Djakarta Raya (PT Bank Pembangunan Daerah Djakarta Raya) No. 30 tanggal 11 April 1961 dibuat oleh dan di hadapan Eliza Pondaag S.H., Notaris di Jakarta, yang telah memperoleh penetapan Menteri Kehakiman Republik Indonesia dengan Surat Keputusan No. J.A.5/31/13 tanggal 11 April 1961 dan telah didaftarkan dalam buku register di Kantor Pengadilan Negeri Jakarta No. 1274 tanggal 26 Juni 1961 serta telah diumumkan dalam Tambahan No. 206 Berita Negara Republik Indonesia No. 41 tanggal 1 Juni 1962.

Dalam rangka penyesuaian ketentuan Undang-Undang Republik Indonesia No. 13 Tahun 1962 tentang Ketentuan-Ketentuan Pokok Bank Pembangunan Daerah dan berdasarkan Peraturan Daerah DKI Jakarta No. 6 Tahun 1978 tanggal 21 Agustus 1978 tentang Bank Pembangunan Daerah Jakarta (BPD Jaya), bentuk Badan Hukum Perusahaan diubah dari Perseroan Terbatas Bank Pembangunan Daerah Jakarta Raya menjadi Bank Pembangunan Daerah DKI Jakarta.

Berdasarkan Peraturan Daerah No. 1 Tahun 1993 tanggal 15 Januari 1993 dilakukan penambahan modal dasar dari sebesar Rp50.000.000.000 menjadi sebesar Rp300.000.000.000.

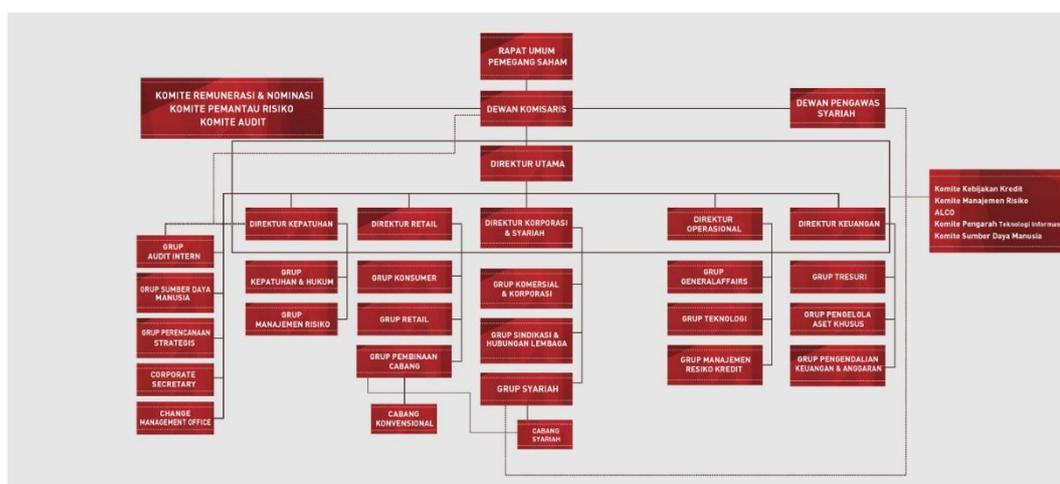
Pada tanggal 1 Februari 1999, Pemerintah Daerah Propinsi DKI Jakarta selaku Pemegang Saham menerbitkan Peraturan Daerah Propinsi DKI Jakarta No. 1 tahun 1999 tentang Perubahan Bentuk Hukum Bank Pembangunan Daerah DKI Jakarta dari Perusahaan Daerah menjadi Perseroan Terbatas Bank Pembangunan Daerah DKI Jakarta, sehingga bentuk Badan Hukum Perusahaan yang semula Perusahaan Daerah (PD) berubah menjadi Perseroan Terbatas (PT) dengan modal dasar sebesar Rp700.000.000.000 sebagaimana tercantum dalam Akta No. 4 tanggal 6 Mei 1999 tentang Akta Pendirian Perseroan Terbatas yang dibuat oleh dan dihadapan Notaris Harun Kamil, S.H., di Jakarta dan telah disahkan oleh Menteri Kehakiman berdasarkan Surat Keputusan No. C-8270.HT.01.01.Th. 99 tanggal 7 Mei 1999 dan diumumkan dalam Berita Negara No. 45, Tambahan No. 3283 tanggal 4 Juni 1999. Dalam rangka penyesuaian ketentuan Undang-Undang Republik Indonesia No. 40 Tahun 2007 tentang Perseroan Terbatas, Bank DKI melakukan perubahan Anggaran Dasar termasuk penambahan modal dasar menjadi Rp1.500.000.000.000 sebagaimana tercantum dalam Akta No. 21 tanggal 12 September 2008 tentang Pernyataan Keputusan Rapat PT. Bank DKI yang dibuat oleh dan dihadapan Notaris Ny. Poerbaningsih Adi Warsito, S.H., Notaris di Jakarta yang telah mendapatkan Persetujuan Menteri Hukum dan Hak Asasi Manusia Republik Indonesia dengan No. AHU-79636.AH.01.02.Tahun 2008 tanggal 29 Oktober 2008. Sebagaimana tercantum dalam Akta No. 09 tanggal 5 November 2012 tentang Pernyataan Keputusan Rapat PT. Bank DKI yang dibuat oleh dan dihadapan Notaris Nanette Cahyanie Handari Adi Warsito, S.H., Notaris di Jakarta, telah dilakukan penambahan modal dasar yang semula Rp1.500.000.000.000 menjadi Rp3.500.000.000.000 dan telah mendapatkan Persetujuan Menteri Hukum dan Hak Asasi Manusia Republik Indonesia dengan Surat Keputusan No. AHU-57968.AH.01.02 tahun 2012 tanggal 13 November 2012. Perubahan modal dasar ini telah didudukkan dalam Peraturan Daerah Provinsi DKI Jakarta No. 8 Tahun 2012

### 2.3.2 Visi dan Misi Bank DKI Jakarta

Visi dan Misi Bank DKI merupakan arah dan tujuan yang hendak dicapai oleh setiap Manajemen dan Karyawan Bank DKI. Pada tahun 2006, seluruh karyawan Bank DKI, bersama dengan manajemen pada saat itu, telah merumuskan visi dan misi serta nilai budaya kerja sebagai upaya perubahan budaya di Bank DKI dan telah ditetapkan dalam Keputusan Direksi No. 156 Tahun 2006 tanggal 11 Desember 2006 tentang Penetapan Visi, Misi dan Nilai-Nilai Budaya Perusahaan PT. Bank DKI. Visi Bank DKI “Menjadi Bank Terbaik Yang Membanggakan” sedangkan Misi Bank DKI “Bank berkinerja unggul, mitra strategis dunia usaha, masyarakat dan andalan pemerintah provinsi DKI Jakarta yang memberi nilai tambah bagi stakeholder melalui pelayanan terpadu dan profesional.

### 2.3.3 Struktur Perusahaan

Adapun Struktur Organisasi pada Bank DKI Jakarta sebagai berikut :

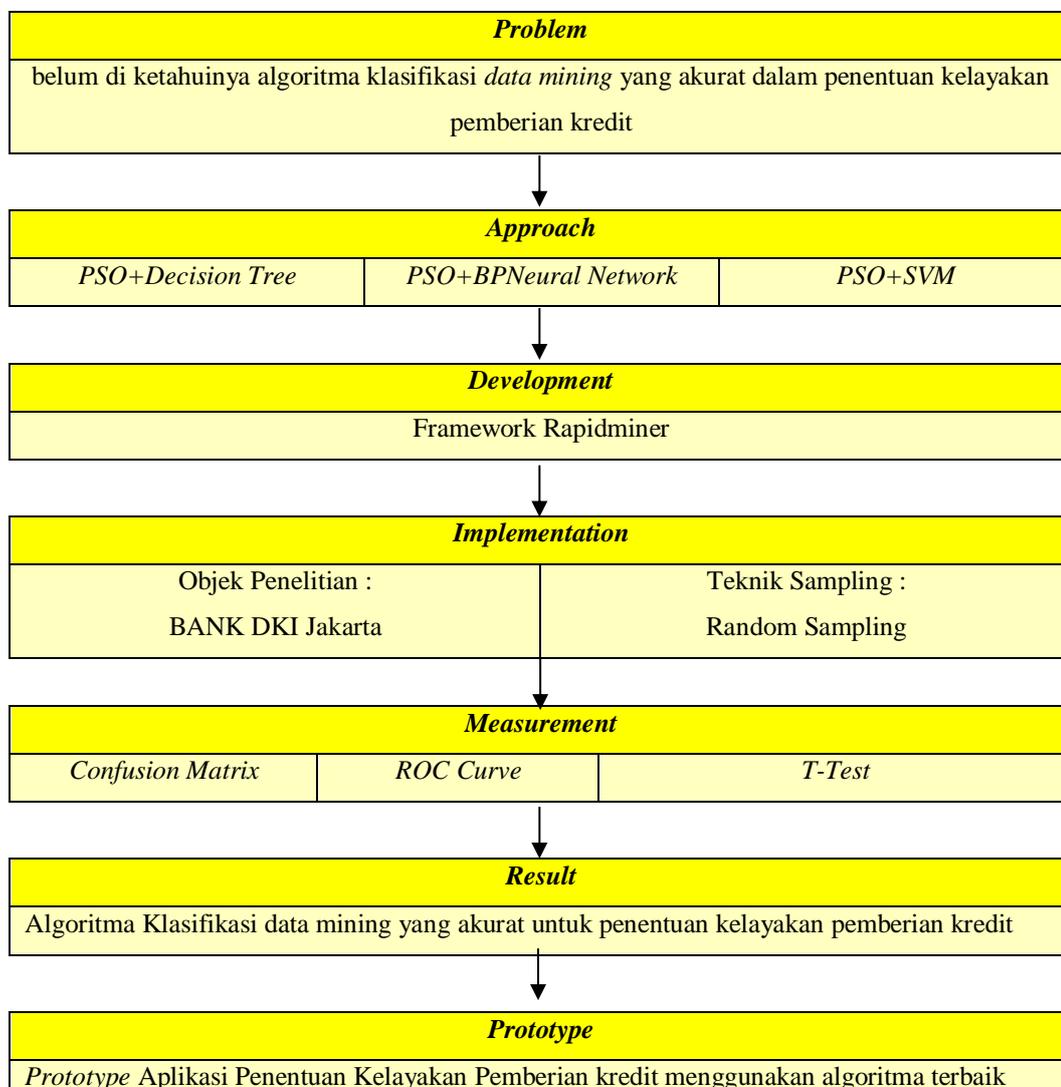


Sumber : Bank DKI Jakarta

**Gambar 2.8 Struktur Organisasi Bank DKI Jakarta**

## 2.4 Kerangka Pemikiran

Penelitian ini dimulai dari kesulitan dari pengklasifikasian antara nasabah baik dan nasabah buruk dalam hal penentuan kelayakan pemberian kredit. berikut merupakan kerangka pemikiran dalam bentuk bagan :



Gambar 2.9 Kerangka Pemikiran

Kerangka pemikiran yang saya ajukan dalam penelitian kali ini dimulai dari penemuan suatu masalah setelah peneliti melakukan studi literatur dengan beberapa jurnal yang terkait dengan masalah kelayakan kredit, masalah tersebut adalah belum di ketahuinya algoritma klasifikasi datamining yang akurat dalam penentuan pemberian kredit. Oleh karena itu peneliti mencoba melakukan eksperimen penelitian dengan mengkomparasi 3 algoritma klasifikasi data mining yaitu *Decision Tree*, *Support Vector Machine*, dan *Neural Network* dengan

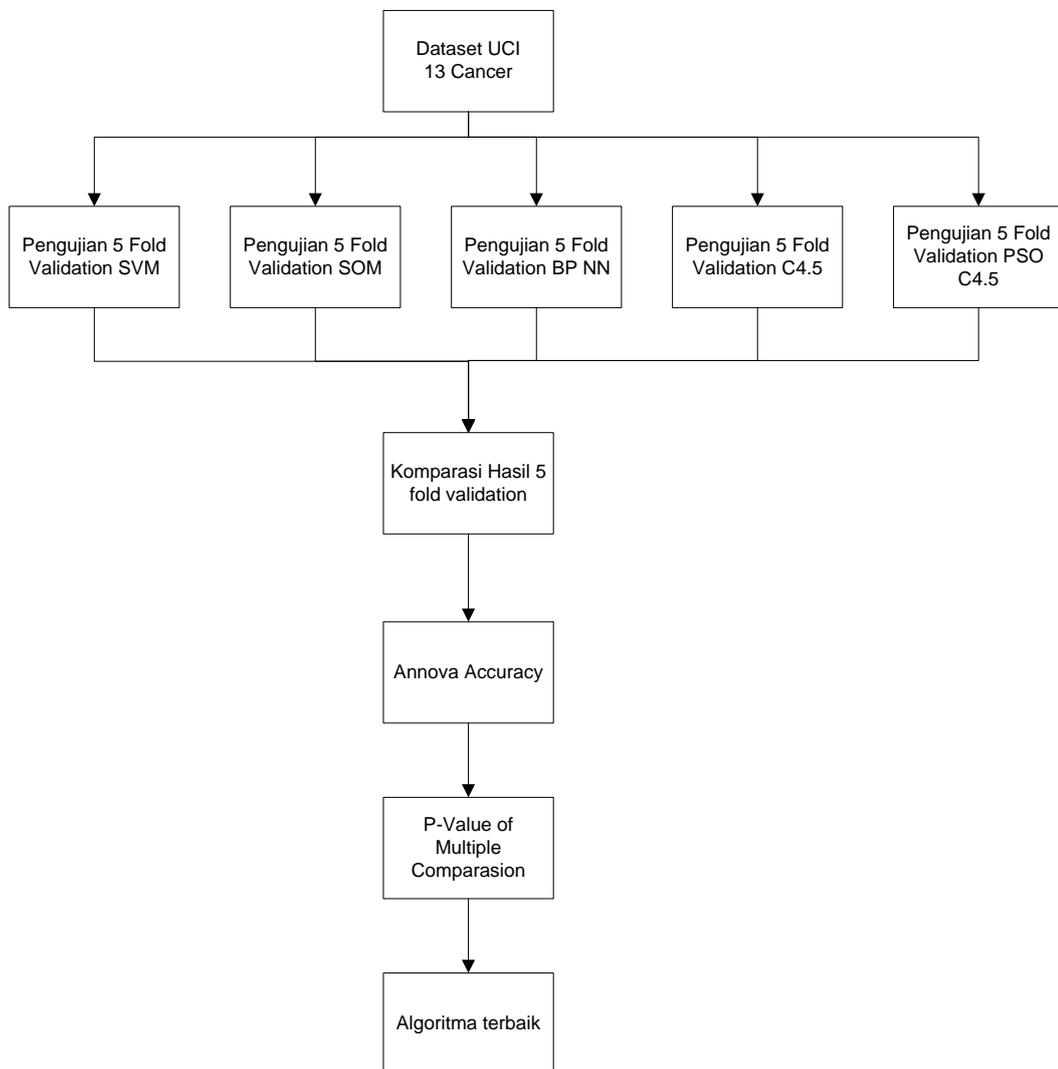
menambahkan algoritma seleksi atribut yaitu *Particle Swarm Optimazation*. Peneliti melakukan ujicoba terhadap ke 3 algoritma tersebut menggunakan software *rapidminer* dengan dataset yang diambil dari Bank DKI Jakarta Cab. Bendungan Hilir. Setelah melakukan ujicoba maka peneliti akan mengevaluasi hasil ujicoba berupa *Confusion Matrix*, *ROC Curve*, dan hasil *T-Test*. Dari hasil evaluasi tersebut maka akan di dapatkan suatu algoritma klasifikasi yang akurat. Selanjutnya peneliti akan membuat sebuah prototype dari algoritma terbaik dan melakukan pengujian terhadap prototype tersebut.

## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

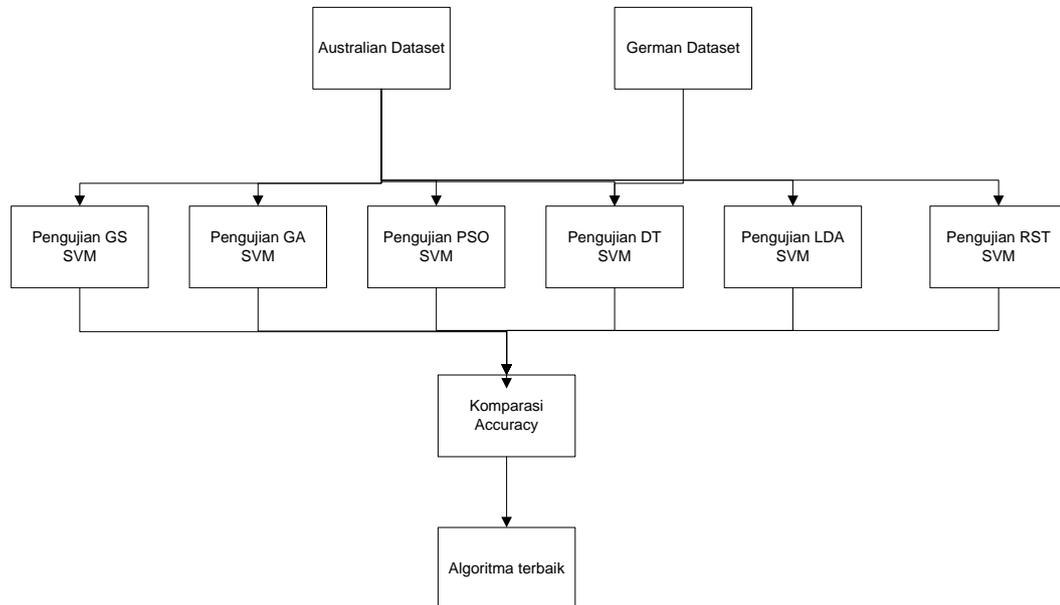
Penelitian dengan model komparasi algoritma sudah dilakukan oleh beberapa peneliti berikut ini merupakan model penelitian yang telah dilakukan oleh beberapa peneliti diantaranya :

Model penelitian yang dilakukan Kun-Huang Chen, Kung-Jeng Wang\*, Kung-Min Wang, Melani-Adrian Angelia sebagai berikut:



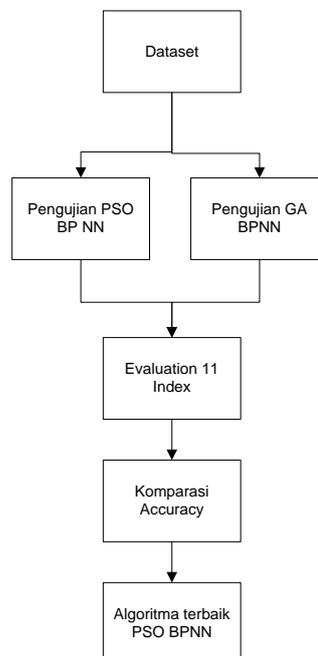
Gambar III.1 Model Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression Data

Model penelitian yang dilakukan Yun Li, Qiu Yancoo, Hua Zhang sebagai berikut:



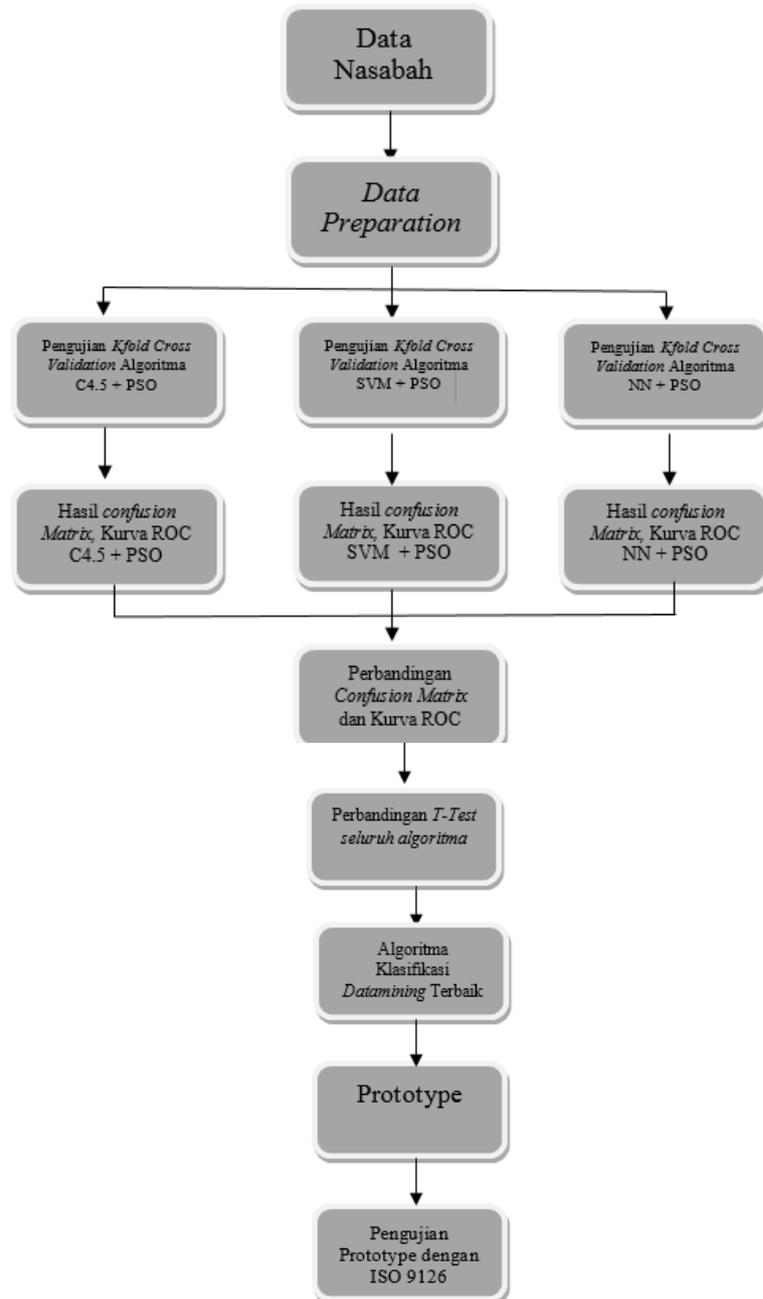
Gambar III. 2 Application of the PSO-SVM model for Credit Scoring

Model Penelitian yang dilakukan Ning Liu, En Jun Xia, Li Yang sebagai berikut :



Gambar III.3 Research and application of PSO-BP Neural Networks in Credit Risk Assessment

Dari beberapa model tersebut maka peneliti menggunakan metode penelitian yang akan digunakan dalam eksperimen komparasi algoritma *Support Vector Machine*, *Decision Tree*, dan *Neural Network* berbasis *Particle Swarm Optimazation* ini dengan menggunakan model penelitian sebagai berikut :



Gambar III.4 Model yang diusulkan

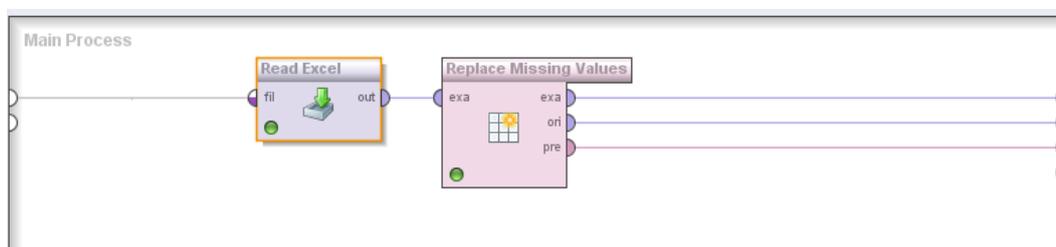
### 3.2 Data Nasabah

Data kredit diambil dari Bank DKI Jakarta Cabang Bendungan Hilir pada tahun 2012 sampai 2013, dimana dari 632 data debitur, ditemukan 97 debitur yang mengalami masalah dalam pembayaran angsuran. Ada beberapa atribut yang digunakan dalam data debitur yaitu akun bank, lama pinjaman, jaminan, jumlah kredit, jumlah angsuran, umur, pekerjaan, masa kerja, gaji, persentase pemotongan gaji, jenis kelamin, status, pinjaman sebelumnya, response. Nilai dari atribut tersebut ada yang merupakan nilai kategorikal diantaranya akun bank, jaminan, pekerjaan, masa kerja, jenis kelamin, status, pinjaman sebelumnya

### 3.3 Data Preparation

Data yang diperoleh untuk penelitian ini sebanyak 695 *record* transaksi kredit baik yang bermasalah maupun yang tidak bermasalah. Untuk mendapatkan data yang berkualitas, beberapa teknik *preprocessing* digunakan, yaitu: (Vecellis, 2009)

1. *data validation*, untuk mengidentifikasi dan menghapus data yang ganjil (*outlier/noise*), data yang tidak konsisten, dan data yang tidak lengkap (*missing value*)



Gambar 3.2 Proses *Replace Missing Value* Pada Rapidminer

Setelah dilakukan proses *replace missing value* terdapat 2 atribut yang memiliki *Missing value* yaitu atribut *response* dan *jaminan*

Role	Name	Type	Statistics	Range	Missings
label	response	binominal	mode = Lancar (535), least = Bermasalah (96)	Lancar (535), Bermasalah (96)	1
regular	akun_bank	binominal	mode = Tidak_Memiliki (319), least = Memiliki (313)	Memiliki (313), Tidak_Memiliki (319)	0
regular	lama_pinjam	integer	avg = 53.750 +/- 24.509	[11.000 ; 120.000]	0
regular	jaminan	binominal	mode = Surat_berharga (594), least = Rumah (35)	Surat_berharga (594), Rumah (35)	3
regular	jumlah_kredit	integer	avg = 70830475.665 +/- 49549850.702	[10000000.000 ; 272422830.000]	0
regular	jumlah_angsuran	integer	avg = 2048268.030 +/- 1173181.430	[380000.000 ; 9357176.000]	0
regular	umur	integer	avg = 44.380 +/- 7.681	[20.000 ; 64.000]	0
regular	pekerjaan	polynomial	mode = Karyawan_Swasta (166), least = Pensiunan (24)	Karyawan_Swasta (166), Guru (159), PNS (119), Pe	0
regular	masa_kerja	integer	avg = 3.777 +/- 0.579	[2.000 ; 4.000]	0
regular	gaji	integer	avg = 5689128.967 +/- 2809813.802	[767158.000 ; 22734069.000]	0
regular	persentase_pemotongan_gaji	numeric	avg = 37.681 +/- 14.236	[9.540 ; 123.170]	0
regular	jenis_kelamin	binominal	mode = Pria (417), least = Wanita (215)	Pria (417), Wanita (215)	0
regular	status	binominal	mode = Menikah (624), least = Lajang (8)	Menikah (624), Lajang (8)	0
regular	pinjaman_sebelumnya	polynomial	mode = lunas (388), least = belum_lunas (7)	lunas (388), Tidak_ada_pinjaman (237), belum_lun	0

Gambar III.5 Hasil Proses *Replace Missing Value* Pada Rapidminer

2. *data integration*, untuk meningkatkan akurasi dan efisiensi algoritma. Data yang digunakan dalam penulisan ini bernilai kategorikal. Untuk model *neural network*, dan *SVM* data ditransformasi ke dalam angka.

Tabel 3.1 dibawah ini menampilkan nama atribut, kategori, dan nilai angka(range), berikut rule nilainya :

**Tabel 3.1 Kategori Atribut**

no	Atribut	Nilai	Keterangan
1	akun_bank	1	Tidak Memiliki akun bank
		2	Memiliki akun bank
2	Jaminan	1	Surat - Surat Berharga
		2	Rumah
		3	Tanah
		4	Lain lain
3	pekerjaan	1	Pensiunan
		2	PNS
		3	Karyawan Swasta
		4	Wiraswasta
		5	guru
4	masa_kerja	1	< 1 tahun
		2	1 - 4 tahun
		3	4 - 7 tahun
		4	> 7tahun
5	Jenis_kelamin	1	Pria
		2	Wanita
6	Status	1	Single
		2	Menikah
		3	Bercerai
7	pinjaman sebelumnya	1	tidak ada pinjaman
		2	ada pinjaman (belum lunas)
		3	ada pinjaman (sudah lunas)
8	Response	1	Lancar
		2	Bermasalah

3. *data transformation* metode *Min Max*, unit ukuran dapat mempengaruhi data, unit yang lebih kecil akan menghasilkan rentang nilai yang besar, data seperti ini sangat di perlukan dalam proses klasifikasi Neural Network dan SVM (Han & Kamber, 2007), berikut merupakan proses transformasi dengan metode min – max :

$$x^1 = \frac{0.8(x-a)}{b-a} + 0.1 \quad (3.1)$$

Keterangan :  $X^1$  = Nilai Transform

X = Nilai Asli

a = Nilai Minimal

b = Nilai Maximal

Berdasarkan sampel acak nilai output (class) kelayakan kredit sebagai berikut :

1. Jumlah Kredit

Diketahui :

Nilai Minimum (a) = 10000000

Nilai Maximum (b) = 2722422830

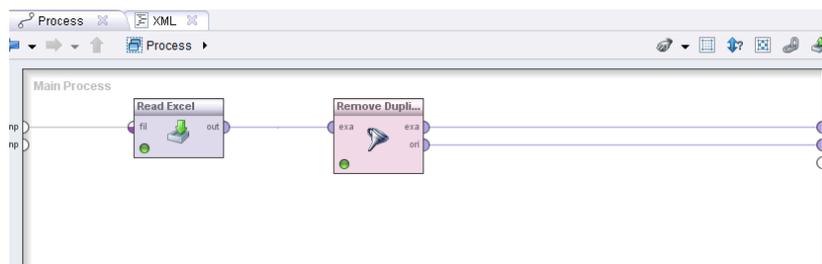
Data yang akan ditransform (x) = 3277778

Dataset yang telah ditransformasikan terdapat pada tabel 3.2 sebagai berikut:

Tabel 3.2 Data Hasil *Transformation Min Max*

	akun_bank	lama_pinjam	jaminan	jumlah_kredit	jumlah_angsuran	umur	pekerjaan	masa_kerja	gaji	persentase_pemotongan_gaji	jenis_kelamin	status	pinjaman_sebelumnya	response
2	2	0,15	1	0,22	0,36	0,32	3	2	0,35	0,32	1	2	3	Lancar
3	2	0,10	1	0,14	0,27	0,74	3	4	0,17	0,64	1	2	3	Bermasalah
4	2	0,10	1	0,11	0,17	0,65	3	4	0,16	0,35	1	2	3	Lancar
5	1	0,28	1	0,19	0,20	0,37	5	4	0,19	0,35	2	2	1	Lancar
6	1	0,20	1	0,15	0,18	0,28	5	4	0,16	0,30	1	2	1	Lancar
7	1	0,28	1	0,15	0,15	0,32	3	3	0,20	0,22	2	2	1	Lancar
8	1	0,28	1	0,15	0,15	0,54	5	4	0,22	0,19	1	2	1	Lancar
9	1	0,20	1	0,15	0,18	0,54	3	4	0,24	0,23	1	2	1	Lancar
10	2	0,20	1	0,28	0,38	0,46	3	4	0,23	0,55	1	2	1	Bermasalah
11	1	0,46	1	0,15	0,15	0,65	5	4	0,25	0,17	2	2	1	Lancar
12	2	0,28	1	0,29	0,31	0,65	5	4	0,25	0,42	2	2	1	Bermasalah
13	1	0,20	1	0,16	0,20	0,48	3	4	0,18	0,38	2	2	1	Bermasalah
14	1	0,20	1	0,29	0,41	0,48	3	4	0,18	0,90	2	2	3	Bermasalah

4. *data size reduction and dicretization*, untuk memperoleh data set dengan jumlah atribut dan *record* yang lebih sedikit tetapi bersifat informatif. Di dalam data *training* yang digunakan dalam penelitian ini, dilakukan seleksi atribut dari 28 atribut menjadi 14 atribut. Selain itu dilakukan pembuangan terhadap data yang terduplikasi seperti gambar dibawah ini



Gambar III.6 Proses *Remove Duplicate* Pada Rapidminer

Setelah dilakukan *remove duplicate* data yang didapat dari kredit sebanyak 715 *record* direduksi dengan menghilangkan duplikasi menjadi 695 *record* untuk data *training* seperti tabel dibawah ini.

Tabel 3.3 yang merupakan Sampel Data Asli.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	akun_bank	lama_pinjam	jaminan	jumlah_kredit	jumlah_angsuran	umur	pekerjaan	masa_kerja	gaji	persentase_pemotongan_gaji	jenis_kelamin	status	pinjaman_sebelumnya	response
1	Memiliki	18	Surat_berharga	50000000	3277778	32	Karyawan_Swasta	2	7702500	42,55	Pria	Menikah	lunas	Lancar
2	Memiliki	11	Surat_berharga	23000000	2320910	55	Karyawan_Swasta	4	2560012	90,66	Pria	Menikah	lunas	Bermasalah
3	Memiliki	11	Surat_berharga	12000000	1210910	50	Karyawan_Swasta	4	2500012	47,3	Pria	Menikah	lunas	Lancar
4	Tidak_Memiliki	36	Surat_berharga	40000000	1511112	35	Guru	4	3251586	46,47	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
5	Tidak_Memiliki	24	Surat_berharga	25000000	1291667	30	Guru	4	2301980	39,72	Pria	Menikah	Tidak_ada_pinjaman	Lancar
6	Tidak_Memiliki	36	Surat_berharga	25000000	944445	32	Karyawan_Swasta	3	3393819	27,83	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
7	Tidak_Memiliki	36	Surat_berharga	25000000	944445	44	Guru	4	4021716	23,48	Pria	Menikah	Tidak_ada_pinjaman	Lancar
8	Tidak_Memiliki	24	Surat_berharga	25000000	1291667	44	Karyawan_Swasta	4	4346089	28,41	Pria	Menikah	Tidak_ada_pinjaman	Lancar
9	Memiliki	24	Surat_berharga	68000000	3523219	40	Karyawan_Swasta	4	4346089	77,5	Pria	Menikah	Tidak_ada_pinjaman	Bermasalah
10	Tidak_Memiliki	60	Surat_berharga	25000000	944445	50	Guru	4	4900595	19,27	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
11	Memiliki	36	Surat_berharga	73000000	2777004	50	Guru	4	4900595	56,67	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
12	Tidak_Memiliki	24	Surat_berharga	29000000	1458334	41	Karyawan_Swasta	4	2948699	50,81	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
13	Tidak_Memiliki	24	Surat_berharga	73000000	3808737	41	Karyawan_Swasta	4	2984700	129,17	Wanita	Menikah	lunas	Bermasalah
14	Memiliki	24	Surat_berharga	28000000	1446667	41	Karyawan_Swasta	4	2948699	49,06	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
15	Memiliki	12	Surat_berharga	20000000	1833334	24	PNS	2	3689571	49,69	Wanita	Menikah	lunas	Bermasalah
16	Memiliki	12	Surat_berharga	19900000	1824167	24	PNS	2	3689572	49,44	Wanita	Menikah	lunas	Lancar
17	Tidak_Memiliki	36	Surat_berharga	20000000	747223	42	Karyawan_Swasta	3	4546912	16,43	Pria	Menikah	Tidak_ada_pinjaman	Lancar
18	Tidak_Memiliki	36	Surat_berharga	113672800	4246943	40	Karyawan_Swasta	3	4546912	93,4	Pria	Menikah	Tidak_ada_pinjaman	Bermasalah
19	Memiliki	60	Surat_berharga	50000000	1270834	32	PNS	2	4288072	29,64	Pria	Menikah	lunas	Lancar
20	Memiliki	60	Surat_berharga	128642160	3269655	32	PNS	2	4288072	76,25	Pria	Menikah	lunas	Bermasalah
21	Memiliki	36	Surat_berharga	38000000	1356389	39	PNS	2	4076200	33,28	Wanita	Menikah	lunas	Lancar
22	Memiliki	48	Surat_berharga	40000000	1183334	39	PNS	2	3690800	32,06	Wanita	Menikah	lunas	Lancar
23	Memiliki	36	Surat_berharga	27000000	963750	39	PNS	2	3993100	24,14	Wanita	Menikah	lunas	Lancar
24	Tidak_Memiliki	36	Surat_berharga	25000000	934028	40	Karyawan_Swasta	2	2603693	35,87	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
25	Tidak_Memiliki	36	Surat_berharga	39000000	1459153	40	Karyawan_Swasta	2	2603693	56,04	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
26	Tidak_Memiliki	36	Surat_berharga	30000000	1120834	39	Karyawan_Swasta	2	2079260	53,91	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
27	Tidak_Memiliki	36	Surat_berharga	25000000	934028	39	Karyawan_Swasta	2	2179260	44,92	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
28	Tidak_Memiliki	36	Surat_berharga	30000000	1120834	48	Karyawan_Swasta	4	2830747	39,59	Pria	Menikah	Tidak_ada_pinjaman	Lancar
29	Tidak_Memiliki	36	Surat_berharga	50000000	1868056	45	Karyawan_Swasta	4	2830777	65,99	Pria	Menikah	Tidak_ada_pinjaman	Bermasalah
30	Tidak_Memiliki	60	Surat_berharga	40000000	1083334	31	Karyawan_Swasta	3	7931401	36,96	Pria	Menikah	Tidak_ada_pinjaman	Lancar

### 3.4 Pengujian Algoritma

Dalam penelitian ini akan dilakukan analisis komparasi menggunakan tiga metode klasifikasi data mining. Metode yang diusulkan untuk pengolahan data debitur kredit adalah penggunaan algoritma C4.5, *neural network* dan *Support Vector Machine* serta di tingkatkan dalam segi seleksi atribut dengan *Particle Swarm Optimazation*, setelah diolah dan menghasilkan model, maka terhadap model yang dihasilkan tersebut dilakukan pengujian menggunakan 90 % data pada k-fold cross validation, sedangkan 10 % dari data akan di pakai untuk testing. Kemudian dilakukan evaluasi dan validasi hasil dengan confusion matrix dan kurva ROC. Tahap selanjutnya adalah memperbandingkan hasil akurasi dan AUC dari setiap model, sehingga diperoleh model dari metode klasifikasi penentuan kelayakan pemberian kredit dengan nilai akurasi dan AUC tertinggi.

Dalam tahapan ini akan dilakukan beberapa langkah-langkah metode yang diusulkan data yaitu seperti berikut:

Hasil pengujian dengan akurasi yang paling tinggi adalah metode yang akan digunakan untuk penentuan kelayakan pemberian kredit ini. Berikut gambaran kateristik dari masing-masing metode:

1. Algoritma C4.5 yaitu salah satu algoritma dalam metode *decision tree* yang merubah data menjadi pohon keputusan menggunakan rumus perhitungan entropi
2. *Neural network* yaitu metode layaknya *neuron* manusia untuk mencari jalur terbaik.
3. *Support Vector Machine* yaitu model untuk menemukan *hyperlane* terbaik yang memisahkan dua buah *class*.
4. *Particle Swarm Optimazation* yaitu pencarian solusi optimal secara global dalam ruang pencarian melalui interaksi individu dalam segerombolan partikel dengan cara melakukan seleksi terhadap atribut yang ada.

sebagai alat bantu dalam penelitian ini, digunakan spesifikasi *software* dan *hardware* terlihat pada tabel 3.4

**Tabel 3.4 Spesifikasi *Hardware* dan *Software***

<i>Software</i>	<i>Hardware</i>
Sistem Operasi : Windows 7 Ultimate	CPU : Intel Core i5
Aplikasi : Rapidminer	Memory : 8 GB
	Hardisk : 320 GB

### 3.4.1 *Cross Validation*

*Cross validation* merupakan cara untuk menemukan parameter terbaik dari satu model dengan menguji besarnya error pada data test. Setiap kelas data set harus diwakili proporsi yang tepat antara data training dan data testing. Data dibagi secara acak pada masing-masing kelas dengan perbandingan yang sama. Seluruh proses training dan testing diulang berkali-kali dengan sampel yang berbeda untuk mengurangi bias yang disebabkan oleh sampel tertentu. Tingkat kesalahan pada iterasi yang berbeda akan dihitung rata-ratanya untuk menghasilkan *error rate*.

*Cross validation* membagi data ke dalam sampel dengan ukuran yang sama. Digunakan  $k-1$  sampel untuk training dan sampel sisanya untuk testing. Teknik tersebut disebut *k-fold validation*. *K-fold validation* merupakan teknik validasi yang membagi data ke dalam  $k$  bagian dan dilakukan proses klasifikasi. Untuk memprediksi tingkat kesalahan, percobaan dilakukan dengan menggunakan 10 kali *cross validation*. Data dibagi secara acak menjadi 10 bagian, dimana setiap kelas diwakili dengan proporsi yang sama. Prosedur di eksekusi sebanyak 10 kali pada pelatihan dengan set yang berbeda. Dengan dataset yang berbeda-beda, ekstensif tes menunjukkan bahwa 10 adalah jumlah yang tepat untuk mendapatkan estimasi kesalahan yang terbaik.

### **3.5 Evaluasi Hasil Pengujian**

Pada tahap ini dilakukan pengujian terhadap model-model yang dikomparasi untuk mendapatkan informasi model yang paling akurat. Evaluasi dan validasi menggunakan metode *cross validation*, *confusion matrix*, dan, kurva ROC

### **3.6 Pengujian T-Test**

Pada tahap ini dilakukan pengujian T-test dengan menggunakan software rapidminer. Pengujian di lakukan untuk mengetahui algoritma yang paling dominan dan signifikan dalam penelitian ini.

### **3.7 Algoritma Terbaik**

Setelah dilakukan tahap perbandingan terhadap *confusion matrix*, dan, kurva ROC serta pengujian *T-Test*, maka akan di dapatkan algoritma yang paling dominan dan akurat dalam penentuan kelayakan kredit

### **3.8 Perancangan Prototype**

Setelah pembentukan model dan dilakukan analisa dan pengukuran pada tahap sebelumnya, selanjutnya pada tahap ini diterapkan model yang paling akurat untuk penentuan kelayakan pemberian kredit.

### **3.9 Pengujian Prototype**

Pada tahapan ini dilakukan pengujian terhadap prototype yang telah di implementasikan dari model algoritma yang paling akurat pada tahapan sebelumnya. Model pengujiannya yaitu menggunakan standar ISO 9126 dengan cara menyebar kuisioner kepada pihak – pihak yang berkompeten di bidang

perkreditan di Bank DKI Jakarta Cab. Bendungan Hilir. Kuisisioner tersebut berisi pertanyaan yang berdasarkan pada *Portability*, *Functionality*, *Reliability*, *Usability*, *Efficiency*, dan *Maintability* dari prototype yang akan di terapkan sebagai sistem informasi penentuan kelayakan kredit.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Pengolahan Data

Setelah melakukan pengumpulan data dari dataset nasabah kredit Bank DKI Jakarta, langkah selanjutnya adalah melakukan pengolahan data. Sehingga dapat menghasilkan perhitungan yang baik. Berikut pengolahan data yang dilakukan.

##### 4.1.2 Penentuan Data Training dan Data Testing

Pada Penelitian ini menggunakan Metode K-fold Cross Validation untuk membagi data Training dan Testing. Dan pada penelitian ini menggunakan 90 % dataset untuk metode pelatihan dan 10 % dataset akan digunakan untuk metode pengujian.

#### 4.2 Pembahasan

Dalam penelitian ini mempunyai tujuan adalah mengkomparasi algoritma klasifikasi C4.5, *Support Vector Machine*, dan *Neural Network* dengan Berbasis *Particle Swarm Optimazation*. Data dianalisa dengan beberapa algoritma klasifikasi untuk menemukan algoritma klasifikasi yang paling akurat.

##### 4.2.1 Model Algoritma Klasifikasi C4.5

Model yang pertama adalah algoritma klasifikasi C4.5, berikut langkah-langkah yang akan dilakukan:

Tabel IV.1 Hasil Pengujian 10 Split Kfol Cross Validation

Kfold Cross Validation ( 10 Split C4.5)									
Split 1	Split 2	Split 3	Split 4	Split 5	Split 6	Split 7	Split 8	Split 9	Split 10
89,22	92,2	88,95	90,55	86,7	89,9	89,94	88,27	86,28	86,27
Average = 88,83									

Dari hasil tersebut maka untuk algoritma C4.5 Menggunakan data Split ke 3 dalam proses pengujian.

1. Menghitung jumlah kasus class LANCAR dan class BERMASALAH dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut di Tabel 3.2. Baris total *entropy* dihitung berdasarkan data training menggunakan Persamaan (2.1) sebagai berikut

$$\text{Entropy Total} = \left( -\frac{535}{632} * \log_2 \left( \frac{535}{632} \right) \right) + \left( -\frac{97}{632} * \log_2 \left( \frac{97}{632} \right) \right)$$

$$\text{Entropy Total} = 0,6184$$

2. Dengan menggunakan persamaan Gain, hitung jumlah kasus dari masing-masing atribut dan hitung Gain. Sebagai contoh untuk atribut Lama Pinjam:

Lama Pinjam :

$$> 54 = \frac{264}{632}$$

$$\leq 54 = \frac{368}{632}$$

Untuk record lama pinjam > 54 terdiri dari 208 Lancar dan 56 Bermasalah, untuk lama pinjam  $\leq 54$  terdiri dari 327 Lancar dan 41 Bermasalah. Dapat dihitung Entropynya sebagai berikut:

$$\text{Lama Pinjam} > 54 = \left( -\frac{208}{264} \log_2 \left( \frac{208}{264} \right) \right) + \left( -\frac{56}{264} \log_2 \left( \frac{56}{264} \right) \right) = 0,7292$$

$$\text{Lama Pinjam} \leq 54 = \left( -\frac{327}{368} \log_2 \left( \frac{327}{368} \right) \right) + \left( -\frac{41}{368} \log_2 \left( \frac{41}{368} \right) \right) = 0,5041$$

$$\text{Lama Pinjam} \leq 54 \ \& \ > 54 = \frac{368}{632} * 0,5041 + \frac{264}{632} * 0,7292$$

$$\text{Entropy Persentase Pemotongan Gaji} = 0,2935 + 0,3046 = 0,5981$$

$$\text{Gain Persentase Pemotongan Gaji} = 0,6184 - 0,5981 = 0,0203$$

Dan untuk selengkapnya dapat dilihat di tabel III.4 dibawah ini :

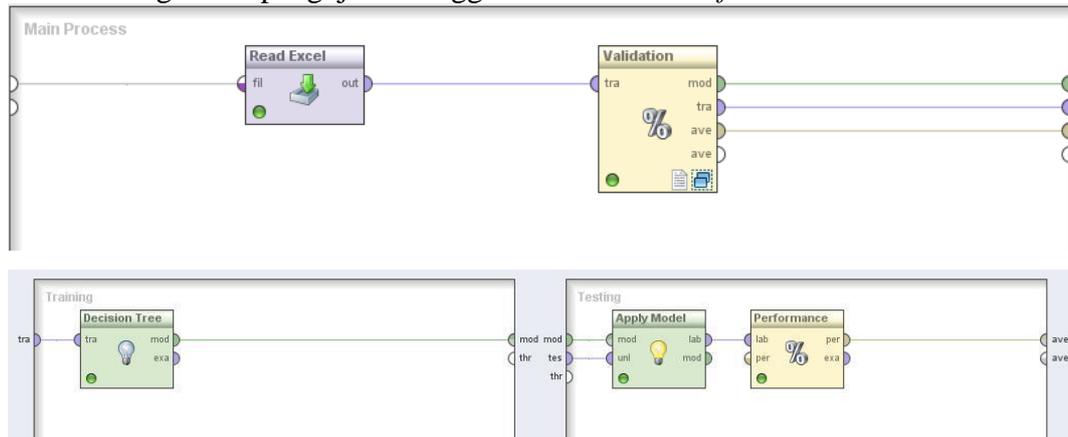
Tabel 4.1 Informasi gain algoritma C4.5

SIMPUL	KASUS	Lancar	Bermasalah	ENTROPY	GAIN
Jumlah Kasus	632	535	97	0,6184	
<b>Akun Bank</b>					<b>0,0119</b>
Tidak Memiliki Akun	218	198	20	0,4422	0,1525
Memiliki Akun Bank	414	337	77	0,693	0,4540
<b>Lama Pinjam</b>					<b>0,0203</b>
>54	264	208	56	0,7292	0,3046
<=54	368	327	41	0,5041	0,2935
<b>Jaminan</b>					<b>0,0027</b>
Surat-Surat Berharga	594	505	89	0,6094	0,5728

Tabel 4.1 Informasi gain algoritma C4.5

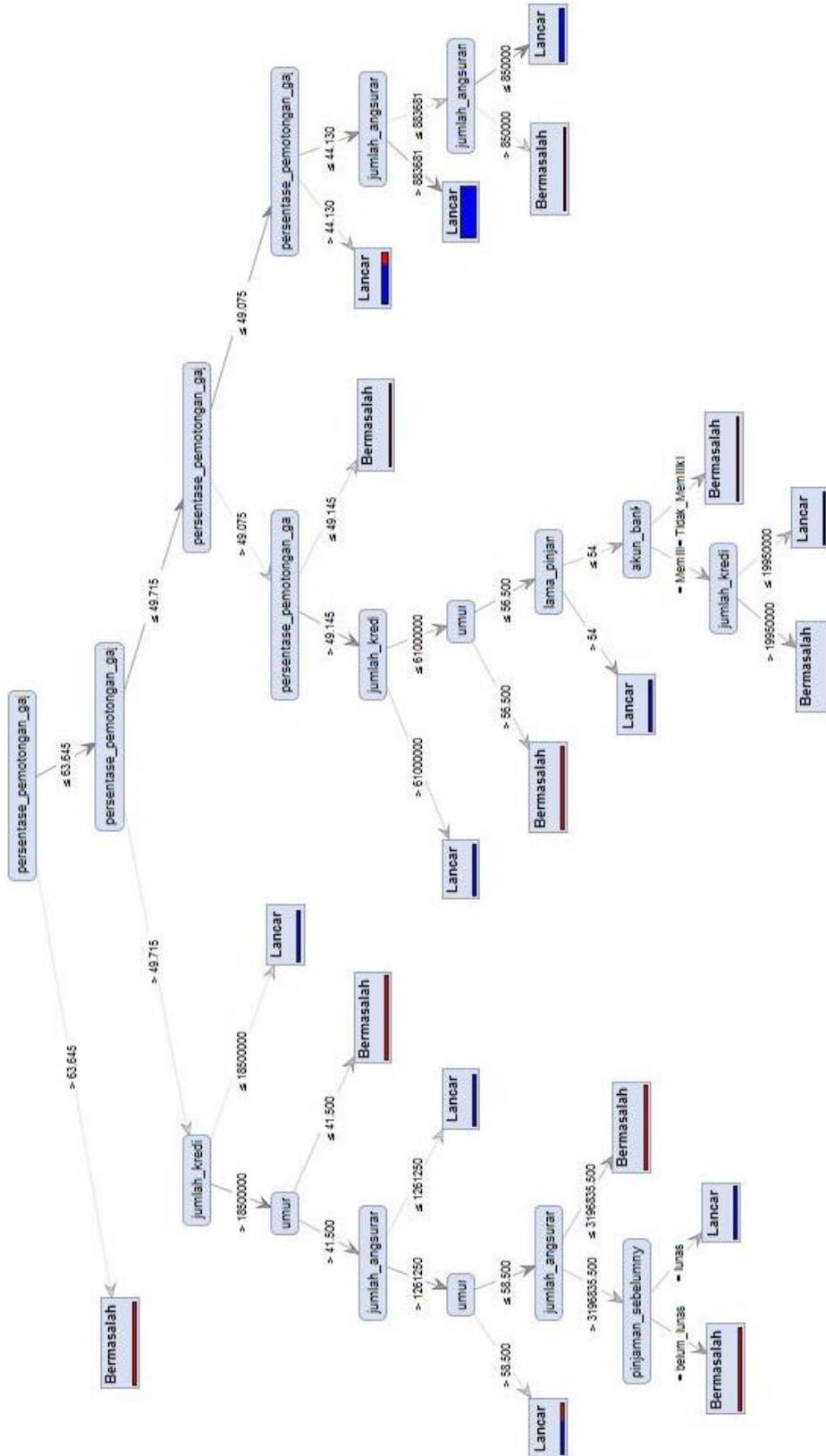
SIMPUL	KASUS	Lancar	Bermasalah	ENTROPY	GAIN
Rumah	35	27	8	0,7755	0,0429
Tanah	3	3	0	0,0000	0,0000
<b>Jumlah Kredit</b>					<b>0,0004</b>
>18500000	606	512	94	0,6224	0,5968
<=18500000	26	23	3	0,5159	0,0212
<b>Jumlah Angsuran</b>					<b>0,0419</b>
>850000	4	2	2	1	0,0063
<=850000	65	53	12	0,6900	0,0710
>883681	117	103	14	0,5284	0,0978
>1261250	346	312	34	0,4635	0,2538
>3196835,500	60	40	20	0,9183	0,0872
>3980938	40	25	15	0,9544	0,0604
<b>Umur</b>					<b>-0,0187</b>
>41,5	391	326	65	0,6825	0,4222
<=41,5	226	197	29	0,5528	0,1977
>58,5	15	12	3	0,7219	0,0171
<b>Pekerjaan</b>					<b>0,0155</b>
Tidak Bekerja	24	13	11	0,9949	0,0378
PNS	119	103	16	0,5695	0,1072
Karyawan Swasta	330	285	45	0,5746	0,3000
Wiraswasta	0	0	0	0,0000	0,0000
Guru	159	134	25	0,6276	0,1579
<b>Persentase Pemotongan Gaji</b>					<b>0,3485</b>
>44.130	110	84	26	0,7889	0,1373
<=44.130	425	423	2	0,0431	0,0290
>49.075	7	3	4	0,9852	0,0109
>49.305	28	15	13	0,9963	0,0441
>49.715	36	10	26	0,8524	0,0486
>63.645	26	0	26	0,00	0,0000
<b>Jenis Kelamin</b>					<b>0,0000</b>
Laki-laki	417	352	65	0,6243	0,4119
Perempuan	215	183	32	0,6069	0,2065
<b>Status</b>					<b>0,0060</b>
Single	8	4	4	1,00	0,0127
Menikah	624	531	93	0,6074	0,5997
Bercerai	0	0	0	0,00	0,0000
<b>Pinjaman Sebelumnya</b>					<b>0,0358</b>
tidak ada pinjaman	237	212	25	0,4861	0,1823
ada pinjaman (belum lunas)	7	0	7	0,00	0,0000
ada pinjaman (sudah lunas)	388	323	65	0,6520	0,4003

Berikut adalah gambar pengujian menggunakan metode *K-fold Cross Validation* :



Gambar 4.1 *K-fold Cross Validation* Algoritma C4.5

Dari tabel 4.1 dapat dilihat nilai gain tertinggi ada pada atribut persentase pemotongan gaji yakni 0.3485 sehingga dapat atribut persentase pemotongan gaji adalah akar dari pohon keputusan, sehingga diperoleh pohon keputusan sebagai berikut :



Gambar 4.2 Pohon Keputusan Data kredit

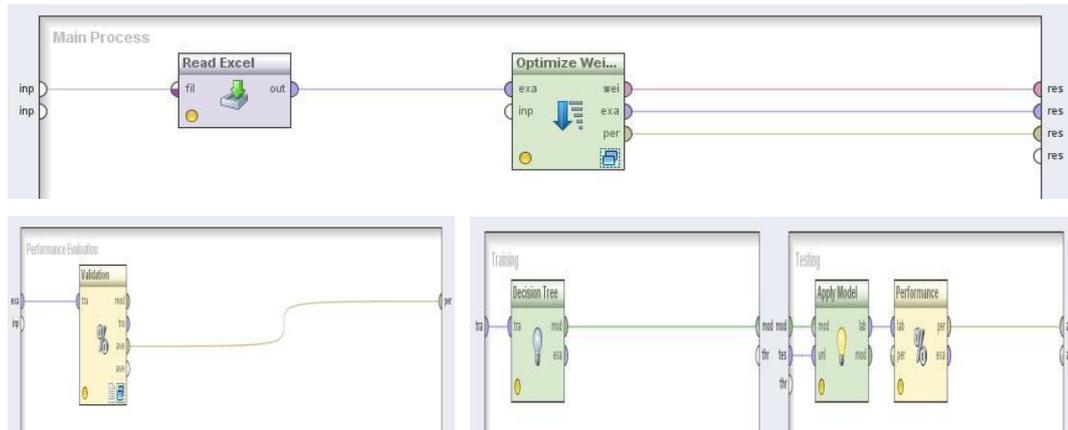
Dari pohon keputusan pada gambar 4.2 didapat *rule* untuk prediksi data kelayakan pemberian kredit, berikut *rule* :

1. R1 : *IF* Persentase Pemotongan Gaji  $> 63.65$  *THEN* Class Bermasalah
2. R2 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $\leq 49.08$  *AND* Persentase Pemotongan Gaji  $\leq 44.13$  *AND* Jumlah angsuran  $\leq 883681$  *AND* Jumlah Angsuran  $\leq 850000$  *Then* Class Lancar
3. R3 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $\leq 49.08$  *AND* Persentase Pemotongan Gaji  $> 44.13$  *Then* Class Lancar.
4. R4 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $\leq 49.08$  *AND* Persentase Pemotongan Gaji  $\leq 44.13$  *AND* Jumlah angsuran  $> 883681$  *Then* Class Lancar
5. R5 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $\leq 49.08$  *AND* Persentase Pemotongan Gaji  $\leq 44.13$  *AND* Jumlah angsuran  $\leq 883681$  *AND* Jumlah Angsuran  $> 850000$  *Then* Class Bermasalah
6. R6 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $> 49.08$  *AND* Jumlah Angsuran  $> 3980938$  *Then* Class Lancar
7. R7 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $> 49.08$  *AND* Jumlah Angsuran  $\leq 3980938$  *AND* Lama Pinjam  $\leq 54$  *Then* Class Bermasalah
8. R8 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $> 49.08$  *AND* Jumlah Angsuran  $\leq 3980938$  *AND* Lama Pinjam  $> 54$  *AND* Persentase Pemotongan Gaji  $\leq 49.3$  *Then* Class Bermasalah
9. R9 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  *AND* Persentase Pemotongan Gaji  $\leq 49.72$  *AND* Persentase Pemotongan Gaji  $> 49.08$  *AND* Jumlah Angsuran  $\leq 3980938$  *AND* Lama Pinjam  $> 54$  *AND* Persentase Pemotongan Gaji  $> 49.3$  *Then* Class Lancar

10. R10 : *mIF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $\leq 18500000$  Then Class Lancar
11. R11 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $\leq 41.5$  Then Class Bermasalah
12. R12 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $> 41.5$  AND Jumlah Angsuran  $\leq 1261250$  Then Class Lancar
13. R13 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $> 41.5$  AND Jumlah Angsuran  $> 1261250$  AND Umur  $\leq 58.5$  AND Jumlah Angsuran  $\leq 3196835.500$  Then Class Bermasalah
14. R14 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $> 41.5$  AND Jumlah Angsuran  $> 1261250$  AND Umur  $\leq 58.5$  AND Jumlah Angsuran  $> 3196835.500$  AND Pinjaman Sebelumnya =belum\_lunas Then Class Bermasalah
15. R15 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $> 41.5$  AND Jumlah Angsuran  $> 1261250$  AND Umur  $\leq 58.5$  AND Jumlah Angsuran  $> 3196835.500$  AND Pinjaman Sebelumnya = lunas Then Class Lancar
16. R16 : *IF* Persentase Pemotongan Gaji  $\leq 63.65$  AND Persentase Pemotongan Gaji  $> 49.72$  AND Jumlah Kredit  $> 18500000$  AND Umur  $> 41.5$  AND Jumlah Angsuran  $> 1261250$  AND Umur  $> 58.5$  Then Class Lancar.

#### **4.2.2 Model Algoritma Klasifikasi C4.5 dengan *Particle Swarm Optimazation***

Pada penelitian prediksi kelayakan pemberian kredit menggunakan algoritma *Decision Tree* berbasis *Particle Swarm Optimization*. *Particle Swarm Optimization* memiliki pencarian kinerja yang lebih baik untuk memecahkan banyak masalah optimasi dengan lebih cepat dan tingkat konvergensi yang stabil.



Gambar 4.3 *K-fold Cross Validation* Algoritma C4.5 berbasis PSO

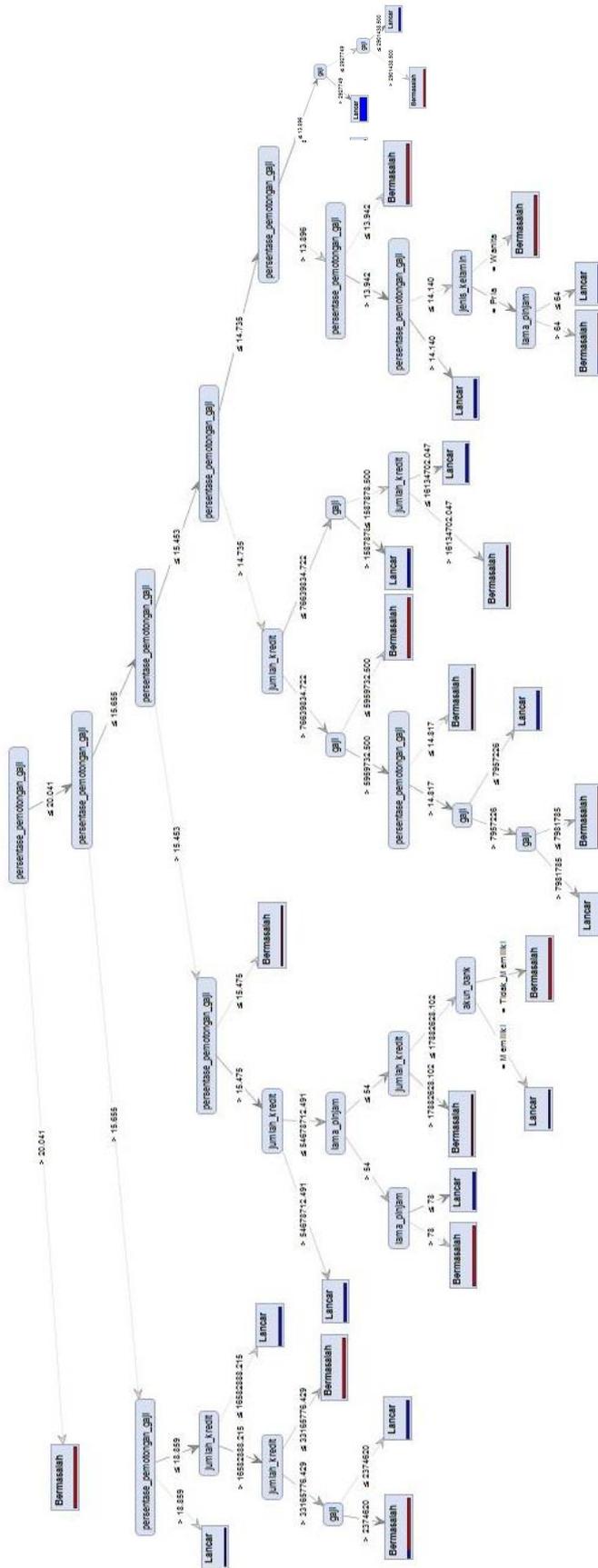
Berdasarkan hasil dari uji coba dengan menggunakan algoritma *decision tree* maka data *training* tersebut akan diseleksi dengan attribute yang digunakan yaitu Akun\_bank, lama\_pinjam, jamina, jumlah\_kredit, jumlah\_angsuran, umur, pekerjaan, masa\_kerja, gaji, persentase\_pemotongan\_gaji, jenis\_kelamin, status, pinjaman\_sebelumnya dan 1 atribut sebagai label yaitu *response*

Tabel 4.2 Bobot Atribut dengan Metode *decision tree* dengan PSO

Attribute	Weight
Akun_bank	0
Lama_pinjam	1
jaminan	0.581
jumlah_kredit	0.958
jumlah_angsuran	1
umur	0
pekerjaan	0
masa_kerja	0
gaji	0.660
persentase_pemotongan_gaji	1
jenis_kelamin	1
status	0
pinjaman_sebelumnya	0

Dari hasil uji coba menggunakan algoritma *decision tree* berbasis *particle swarm optimization* diperoleh beberapa atribut-atribut yang berpengaruh terhadap bobot

atribut yaitu: lama\_pinjam, jaminan, jumlah\_kredit, jumlah\_angsuran, gaji, persentase\_pemotongan\_gaji, jenis\_kelamin. Berdasarkan hasil uji coba akun\_bank, umur, pekerjaan, masa\_kerja, status, pinjaman sebelumnya tidak berpengaruh terhadap bobot attribute, dari bobot yang telah di optimasi dengan *Particle Swarm Optimazation* maka terbentuk pohon keputusan sebagai berikut :



Gambar 4.4 Decision Tree C4.5 berbasis PSO

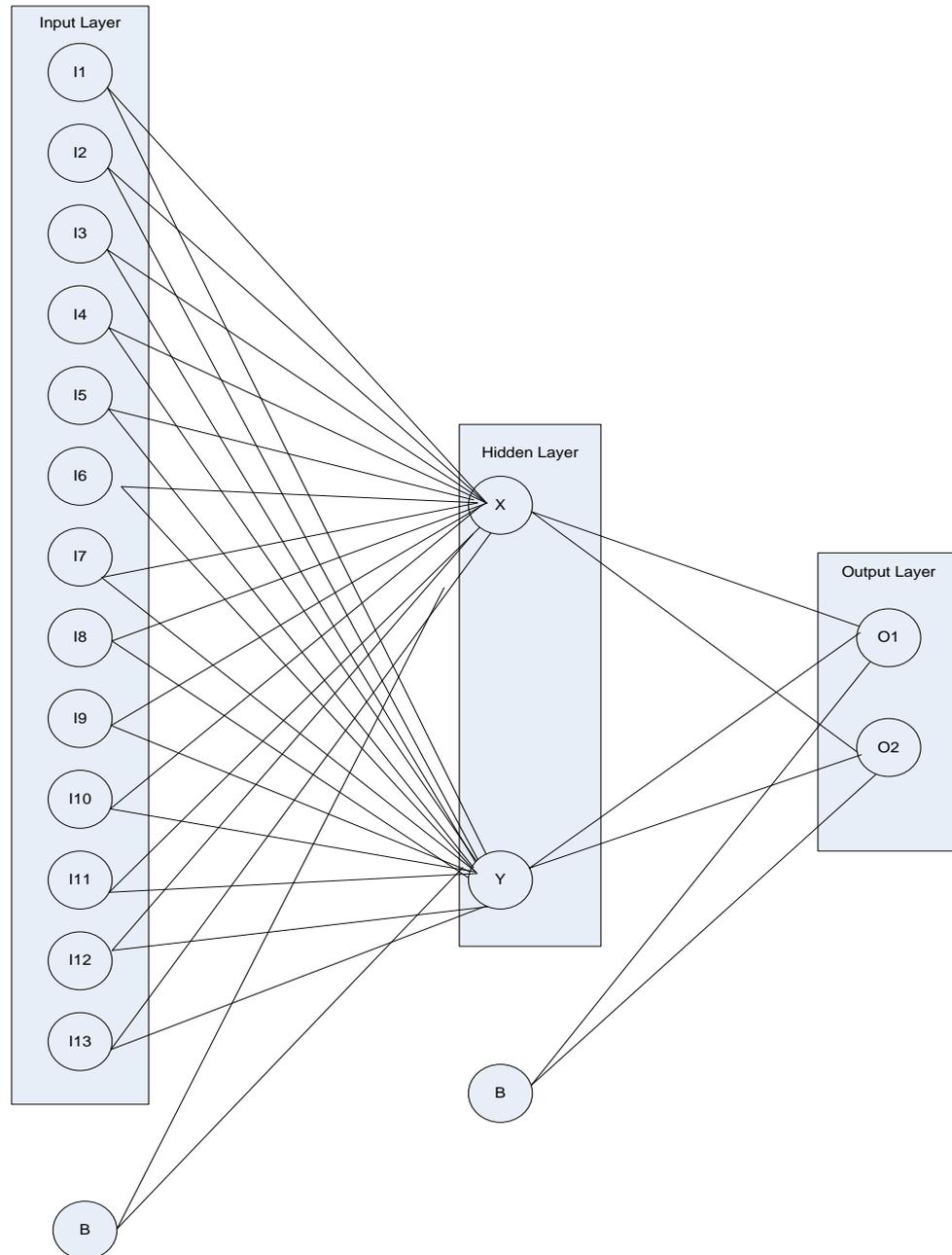
Setelah dilakukan uji coba dengan metode algoritma *decision tree* berbasis PSO maka didapatkan hasil Accuracy sebesar 94,46% dan *Area Under Control* (AUC) sebesar 0.937.

#### **4.2.3 Model Algoritma Klasifikasi *Neural Network***

Algoritma *neural network* adalah algoritma yang didesain untuk operasi pada *feed forward multi layer* dan untuk pelatihan *supervised*, dengan dideksripsikan yaitu: jaringan yang diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit yang ada pada lapisan terluar.

Pembuatan model neural network akan dilakukan pada dataset Kelayakan Kredit. Terdapat 13 atribut dari Kelayakan Kredit dengan kelas Lancar dan Bermasalah.

Dari dataset yang ada, penelitian ini mengambil salah satu eksperimen untuk Kelayakan Kredit dengan menggunakan Learning rate 0.3 dan Momentum 0.2 yang terdiri dari 13 simpul yaitu 1,2,3,4,5,6,7,8,9,10,11,12, dan 13 yang merupakan atribut predictor. Hidden layer sebanyak 2 simpul, yaitu X dan Y dan memiliki output layer terdiri dari 2 simpul yaitu O yang mempresentasikan Output atau class.



**Gambar 4.5** Arsitektur Neural Network

Langkah – langkah dalam perhitungan training neural network (Backpropagation) :

1. Inisialisasi bobot awal secara acak.

Nilai input awal tergambar pada tabel 4.3 dimana tiap simpul di input layer diambil dari dataset Kelayakan Kredit. Terdapat 13 input layer yang menggambarkan 13 atribut yang dimiliki kelayakan kredit dan 1 keluaran yaitu nilai hasil kelayakan kredit. Sedangkan nilai bobot awal untuk input layer dan hidden layer pada tabel 4.4 dimana diberi random -100 s/d 100 pada setiap simpulnya. begitu pula di tabel 4.5 adalah simpul bias yang terhubung dengan hidden layer dan output layer.

**Tabel 4.3 Tabel Nilai Input Awal**

$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$I_8$	$I_9$	$I_{10}$
2	0,15	1	0,22	0,36	0,32	3	2	0,35	0,32

$I_{11}$	$I_{12}$	$I_{13}$	Output /Class
1	2	3	Lancar

Dari tabel diatas, nilai  $I_1$  adalah nilai input pada akun bank di record pertama sebesar 2

**Tabel 4.4 Tabel Nilai Bobot Awal**

$W_{1x}$	$W_{2x}$	$W_{3x}$	$W_{4x}$	$W_{5x}$	$W_{6x}$	$W_{7x}$	$W_{8x}$	$W_{9x}$	$W_{10x}$
-6,593	16,365	-3,878	-11,794	-4,873	4,049	4,007	0,798	17,294	-30,384
$W_{11x}$	$W_{12x}$	$W_{13x}$	$W_{1y}$	$W_{2y}$	$W_{3y}$	$W_{4y}$	$W_{5y}$	$W_{6y}$	$W_{7y}$
-2,844	-5,316	7,902	0,841	-11,238	9,474	0,174	-6,986	1,087	0,494
$W_{8y}$	$W_{9y}$	$W_{10y}$	$W_{11y}$	$W_{12y}$	$W_{13y}$	$W_{x01}$	$W_{x02}$	$W_{y01}$	$W_{y02}$
-3,193	19,948	-59,373	1,6	-2,004	-0,459	12,638	-12,638	14,817	-14,817

Dari tabel diatas  $W_{1x}$  adalah nilai simpul dari Input Layer atribut pertama ke Hidden Layer X sebesar -6,593

**Tabel 4.5 Tabel Nilai Bias awal**

$B_x$	$B_y$	$B_{o1}$	$B_{o2}$
-0,132	-1,206	-13,247	13,247

Dalam tabel 4.5,  $B_x$  merupakan nilai simpul bias dari bias input ke Hidden Layer X sebesar - 0,132

1. Menghitung Input, Output dan Error

$$=(I_1 * W_{1x}) + (I_2 * W_{2x}) + (I_3 * W_{3x}) + (I_4 * W_{4x}) + (I_5 * W_{5x}) + (I_6 * W_{6x}) + (I_7 * W_{7x}) + (I_8 * W_{8x}) + (I_9 * W_{9x}) + (I_{10} * W_{10x}) + (I_{11} * W_{11x}) + (I_{12} * W_{12x}) + (I_{13} * W_{13x}) + B_x$$

$$= (2 * -6.593) + (0.15 * 16.365) + (1 * -3.878) + (0.22 * -11.794) + (0.36 * -4.873) + (0.32 * 4.049) + (3 * 4.007) + (2 * 0.798) + (0.35 * 17.294) + (0.32 * -30.384) + (1 * -2.844) + (2 * -5.316) + (3 * 7.902) + (-0.312)$$

$$=(-13186) + 2.45475 + (-3.878) + (-2.59468) + (-1.75428) + 1.29568 + 12021 + 1.596 + 6.0529 + (-9.72288) + (-2.844) + (-10632) + 23706 + (-0.312)$$

$$= \mathbf{2,405172421}$$

$$\mathbf{Output_x} = 1/(1 + e^{2,405172421}) = \mathbf{0.082779123}$$

$$\mathbf{Input_y} = \mathbf{-14.5688954}$$

$$\mathbf{Output_y} = 1/(1 + e^{-14.5688954}) = \mathbf{0.999999539}$$

$$\begin{aligned} \mathbf{Input_{o1}} &= (\mathbf{Output_x} * W_{xo1}) + (\mathbf{Output_y} * W_{yo1}) + B_{o1} \\ &= (0.082779123 * 12.638) + (0.999999539 * 14.817) + (-13.247) \end{aligned}$$

$$= \mathbf{2.61615558}$$

$$\mathbf{Output_{o1}} = 1/(1 + e^{2.61615558}) = \mathbf{0.068105884}$$

$$\mathbf{Input_{o2}} = \mathbf{-2.61615558}$$

$$\mathbf{Output_{o2}} = 1/(1 + e^{-2.61615558}) = \mathbf{0.931894116}$$

$$\begin{aligned} \mathbf{Error_{o1}} &= \mathbf{Output_{o1}} * (1 - \mathbf{Output_{o1}}) * (\mathbf{Output_{target}} - \mathbf{Output_{o1}}) \\ &= 0.068105884 * (1 - 0.068105884) * (1 - 0.068105884) \end{aligned}$$

$$= \mathbf{0.05914496424}$$

$$\mathbf{Error_{o2}} = \mathbf{0.00432250832}$$

$$\begin{aligned} \mathbf{Error_x} &= \mathbf{Output_x} * (1 - \mathbf{Output_x}) * \mathbf{Error_{o1}} * W_{xo1} \\ &= \mathbf{0.05675326831} \end{aligned}$$

Penelitian ini melakukan uji coba pada nilai *training cycles* ditentukan dengan cara memasukkan nilai *range* dari 100 sampai dengan 1500 untuk *training cycles*, serta nilai 0.3 untuk *learning rate* dan 0.2 untuk *momentum* dengan

menggunakan data training. Berikut ini adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *training cycles*:

Tabel 4.6 Penelitian Penentuan Training Cycles

Training Cycles	Learning Rate	Momentum	Accuracy	AUC
100	0,3	0,2	89,55	0,938
200	0,3	0,2	89,7	0,938
300	0,3	0,2	90,97	0,942
400	0,3	0,2	90,97	0,941
500	0,3	0,2	92,23	0,941
600	0,3	0,2	92,71	0,942
700	0,3	0,2	92,77	0,945
800	0,3	0,2	93,03	0,945
900	0,3	0,2	93,18	0,943
1000	0,3	0,2	93,03	0,945
1100	0,3	0,2	93,18	0,945
1200	0,3	0,2	93,02	0,944
1300	0,3	0,2	93,65	0,943
1400	0,3	0,2	93,34	0,943
1500	0,3	0,2	93,65	0,945

Berdasarkan hasil uji coba pada nilai *training cycles* dipilih berdasarkan nilai akurasi dan nilai *Area Under Curve* (AUC) terbesar yang dihasilkan, dengan nilai 1500 pada *training Cycles*. Kemudian nilai tersebut akan digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Learning Rate*.

Nilai *Learning Rate* ditentukan dengan cara melakukan dengan uji coba memasukkan nilai dengan range 0.1 sampai dengan 0.9. Nilai *Training cycles* dipilih dari percobaan sebelumnya yaitu 1500, sedangkan 0.2 digunakan untuk nilai momentum. Berikut adalah hasil dari eksperimen yang telah dilakukan untuk menentukan nilai *Learning Rate*.

Tabel 4.7 Penelitian Penentuan Nilai *Learning Rate*

Learning Rate	Training Cycles	Momentum	Accuracy	AUC
0,1	1500	0,2	92,34	0,948

0,2	1500	0,2	92,65	0,947
0,3	1500	0,2	92,65	0,945
0,4	1500	0,2	93,29	0,949
0,5	1500	0,2	92,97	0,951
0,6	1500	0,2	92,5	0,943
0,7	1500	0,2	92,66	0,940
0,8	1500	0,2	93,44	0,957
0,9	1500	0,2	92,96	0,950

Berdasarkan hasil uji coba pada nilai *Learning Rate* dipilih berdasarkan nilai akurasi dan nilai *Area Under Curve* (AUC) terbesar yang dihasilkan, dengan nilai 0.8 pada *Learning Rate* dan 200 pada *training Cycles*. Kemudian nilai tersebut akan digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Momentum*.

Nilai momentum ditentukan dengan cara melakukan eksperimen dengan cara memasukkan nilai training cycles 1500 dan *learning rate* 0.8, dan memasukkan nilai momentum dengan *range* 0.1 sampai dengan 0.9. Dengan nilai *training cycles* 1500 dan *learning rate* 0.8 berdasarkan percobaan sebelumnya. Berikut adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai momentum:

Tabel 4.8 Penelitian Penentuan Nilai Momentum

Momentum	Training Cycles	Learning Rate	Accuracy	AUC
0,0	1500	0,8	92,34	0,930
0,1	1500	0,8	92,81	0,947
0,2	1500	0,8	93,44	0,957
0,3	1500	0,8	93,6	0,959
0,4	1500	0,8	92,02	0,926
0,5	1500	0,8	91,71	0,937
0,6	1500	0,8	91,7	0,928
0,7	1500	0,8	89,66	0,896
0,8	1500	0,8	88,22	0,772
0,9	1500	0,8	92,34	0,795

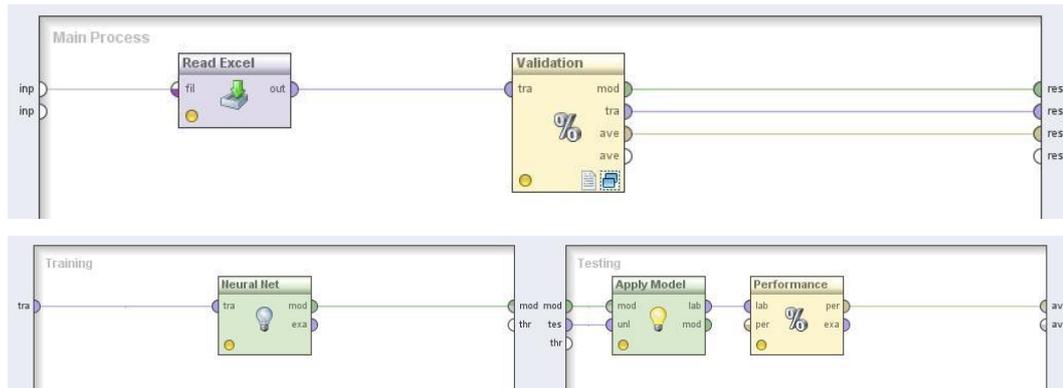
Berdasarkan hasil uji coba diatas, maka untuk parameter *neural network* dipilih nilai 1500 untuk nilai *training cycles*, 0.8 untuk nilai *learning rate* dan 0.3 untuk nilai momentum.

Langkah berikutnya adalah menentukan jumlah *hidden layer* dan *neuron sizes* untuk mencari nilai akurasi dan AUC yang paling tinggi. Pada jumlah hidden layer sebanyak 1, dilakukan percobaan pada *neuron size* dengan *range* 1 sampai dengan sise 25:

Tabel 4.9 Percobaan menentukan *Neuron Size*

Hidden Layer	Accuracy	AUC
1	88,99	0,912
2	88,52	0,9
3	86,93	0,898
4	90,24	0,896
5	90,26	0,906
6	91,06	0,919
7	91,05	0,927
8	92,15	0,924
9	93,1	0,929
10	92	0,918
11	92,15	0,912
12	92,31	0,914
13	91,85	0,914
14	92,47	0,907
15	92,15	0,913
16	92,95	0,929
17	93,1	0,91
18	93,26	0,925
19	93,73	0,914
20	93,58	0,926
21	92,95	0,917
22	93,83	0,944
23	92,94	0,921
24	93,42	0,919
25	93,41	0,929

Berdasarkan data penelitian diatas maka dapat terlihat bahwa dengan satu *hidden layer* dengan *neuron size* 22, mendapatkan nilai accuracy sebesar 93,83% dan nilai AUC sebesar 0.944

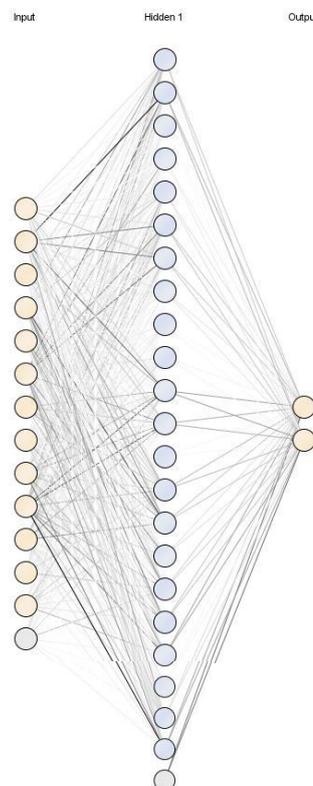


Gambar 4.6 Desain model algoritma *neural network*

Pada desain model algoritma *neural network* diatas dilakukan proses training model dengan memberikan:

1. *Hidden Layer Size* : 22
2. *Training Cycles* : 1500
3. *Learning Rate* : 0.8
4. *Momentum* : 0.3

Dari eksperimen terbaik diatas maka didapatkan arsitektur *neural network* seperti pada gambar 4.6 :



Gambar 4.7 Arsitektur model algoritma *neural network*

Dengan nilai setiap weight model algoritma neural network dapat dilihat pada tabel 4.10 untuk nilai *hidden layer* dan table 4.11. untuk nilai *output layer*

Tabel 4.10 Nilai bobot *hidden layer*

Node	Hidden Layer						
	1	2	3	4	5	6	7
Akun_bank	-11.681	-5,238	-1,775	-1,111	-3,754	-2,237	-3,577
Lama_pinjam	-17.399	-4,572	-0,057	-0,009	-9,240	-5,180	9,585
jaminan	-0,473	-2,921	1,011	0,978	4,657	2,630	-0,078
jumlah_kredit	-10,381	6,005	-0,328	2,430	-16,243	9,543	0,291
jumlah_angsuran	-5,365	4,188	-0,074	1,430	-9,890	5,083	-0,808
umur	23,483	-18,311	2,322	3,644	-7,205	40,173	-9,950
pekerjaan	-13,743	-0,611	-0,241	0,764	-0,956	-10,004	-0,877
masa_kerja	-1,596	0,839	-1,600	-1,180	-4,002	-2,119	-1,213
gaji	7,588	-4,174	2,012	3,299	0,216	14,681	2,514
persentase_pemotongan_gaji	-19,776	15,902	-3,348	-1,579	-14,771	-3,527	-7,388
jenis_kelamin	13,982	-3,548	-0,018	1,035	-3,827	10,531	-2,207
status	-2,791	1,670	-0,842	-1,033	-4,406	-2,505	-0,690
pinjaman_sebelumnya	0,843	0,250	-0,843	0,375	-0,558	2,238	-2,643
bias	-2,893	1,836	-0,860	-1,148	-4,487	-2,588	-0,783

Tabel 4.10 Nilai bobot *hidden layer* (lanjutan)

Node	Hidden Layer							
	8	9	10	11	12	13	14	
Akun_bank	-0,728	-1,734	-2,089	-7,637	-3,289	-0,597	-5,252	
Lama_pinjam	9,385	-0,214	-0,402	-20,615	-10,308	6,489	-37,109	
jaminan	-0,392	0,676	0,905	7,992	0,444	0,077	5,964	
jumlah_kredit	6,715	-0,507	-0,744	-2,205	-1,600	4,200	-7,886	
jumlah_angsuran	2,362	-0,267	-0,251	1,828	2,852	1,596	-5,988	
umur	-11,840	4,128	5,011	8,736	-18,153	-7,240	-16,369	
pekerjaan	1,444	-0,194	-0,871	10,178	0,886	-0,207	-9,442	
masa_kerja	1,955	-1,647	-3,035	2,846	-0,225	0,937	-4,148	
gaji	3,050	1,910	2,235	11,998	18,292	2,476	-7,825	
persentase_pemotongan_gaji	-2,369	-3,863	-5,017	-22,045	-25,885	-1,496	14,636	
jenis_kelamin	-2,552	-0,150	-1,011	-4,061	1,255	-3,199	2,083	
status	0,162	-0,434	-0,536	-2,983	2,632	-0,414	-4,474	
pinjaman_sebelumnya	-0,758	-0,595	-0,944	8,229	0,415	-0,620	4,202	
bias	0,039	-0,487	-0,502	-3,096	2,536	-0,517	-4,325	
Node	Hidden Layer							
	15	16	17	18	19	20	21	22
Akun_bank	-7,987	-2,644	1,742	-0,941	-2,036	-0,880	-1,366	6,139
Lama_pinjam	10,992	-1,167	-0,656	18,547	8,588	0,036	2,759	8,674
jaminan	-1,264	1,907	5,369	0,901	-3,233	1,108	1,682	6,879
jumlah_kredit	23,306	-3,442	-10,869	15,636	-2,114	1,248	-0,122	-8,456
jumlah_angsuran	10,598	-0,763	-6,575	3,978	-4,003	0,951	0,120	-5,653
umur	-7,927	1,624	0,828	-8,267	-15,187	0,480	-1,463	-12,820
pekerjaan	-12,803	-11,251	3,065	11,237	-1,868	0,703	-1,320	-1,362
masa_kerja	1,384	-2,034	-1,873	-1,124	3,347	-1,175	-3,091	-6,397
gaji	8,389	-2,887	0,796	6,671	9,604	1,786	4,009	13,610
persentase_pemotongan_gaji	-4,565	7,058	-11,704	4,219	-20,188	-0,113	-7,213	-34,355
jenis_kelamin	-15,429	5,036	-5,811	-5,109	1,932	0,526	0,049	1,585
status	3,436	-2,470	-8,233	-0,979	-1,156	-1,137	-1,530	-1,539
pinjaman_sebelumnya	-6,859	3,532	2,626	2,235	5,493	-0,220	-0,449	-3,167
bias	3,292	-2,558	-8,327	-1,142	-1,141	-1,227	-1,601	-1,694

Hasil perhitungan akhir backpropagation fungsi aktivasi untuk simpul pada *hidden layer* terdapat pada Tabel 4.10. Kolom pertama pada Tabel 4.10 merupakan atribut yang dinyatakan berupa simpul pada *input layer* seperti pada Gambar 4.7. Sedangkan kolom satu sampai dua puluh dua mewakili jumlah simpul pada *hidden layer*.

Tabel 4.11 Nilai Bobot Akhir untuk output layer

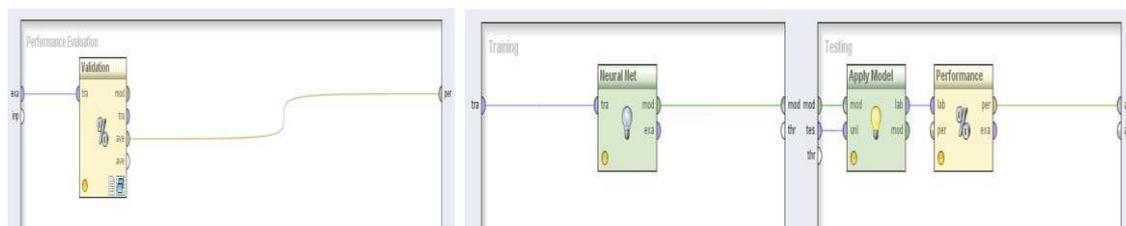
Class	Output Sigmoid							
	1	2	3	4	5	6	7	8
Lancar	9,863	-12,699	1,102	1,132	8,963	6,538	3,451	-5,479
Bermasalah	-9,865	12,698	-1,141	-1,155	-8,964	-6,537	-3,449	5,478
Class	Output Sigmoid							
	9	10	11	12	13	14	15	16
Lancar	1,298	1,762	11,106	9,183	-3,021	-12,638	11,268	-5,173
Bermasalah	-1,301	-1,742	-11,108	-9,182	3,022	12,639	-11,270	5,172
Class	Output Sigmoid							Thresold
	17	18	19	20	21	22		
Lancar	6,346	7,831	8,437	0,327	1,224	9,233	-15,891	
Bermasalah	-6,348	-7,830	-8,436	-0,300	-1,281	-9,232	15,893	

Tabel 4.11. adalah nilai akhir fungsi aktivasi pada *output layer*. Kolom pertama pada Tabel 4.11 menyatakan *class*, yaitu atribut kelas yang dinyatakan dengan simpul pada *output layer* seperti pada gambar 4.7. Nilai yang terdapat pada kolom berlabel angka satu sampai dua puluh dua adalah nilai bias terbaru yang terdapat pada relasi antara simpul pada *hidden layer* dan simpul pada *output layer*.

#### 4.2.4 Model Algoritma Klasifikasi *Neural Network* dengan *Particle Swarm Optimization*

Pada penelitian prediksi kelayakan pemberian kredit menggunakan algoritma *neural network* berbasis *Particle Swarm Optimization*. *Particle Swarm Optimization* memiliki pencarian kinerja yang lebih baik untuk memecahkan banyak masalah optimasi dengan lebih cepat dan tingkat konvergensi yang stabil





Gambar 4.8 Desain model algoritma *neural network* berbasis *Particle Swarm Optimization*

Berdasarkan hasil dari uji coba dengan menggunakan algoritma *neural network* maka data *training* tersebut akan diseleksi dengan attribute yang digunakan yaitu Akun\_bank, lama\_pinjam, jamina, jumlah\_kredit, jumlah\_angsuran, umur, pekerjaan, masa\_kerja, gaji, persentase\_pemotongan\_gaji, jenis\_kelamin, status, pinjaman\_sebelumnya dan 1 atribut sebagai label yaitu *response*.

Tabel 4.12 Bobot Atribut dengan Metode *Neural Network* dengan PSO

Attribute	Weight
Akun_bank	0.401
Lama_pinjam	0.189
jaminan	0.029
jumlah_kredit	0.142
jumlah_angsuran	0.077
umur	0.265
pekerjaan	0.280
masa_kerja	0.765
gaji	0.399
persentase_pemotongan_gaji	0.546
jenis_kelamin	0.503
status	0.769
pinjaman_sebelumnya	0.901

Dari hasil uji coba menggunakan algoritma *neural network* berbasis *particle swarm optimization* diperoleh beberapa atribut-atribut yang berpengaruh terhadap bobot atribut yaitu: akun\_bank, lama\_pinjam, jaminan, umur, pekerjaan, masa\_kerja, gaji,

persentase\_pemotongan\_gaji, jenis\_kelamin, jumlah\_kredit, jumlah\_angsuran, status, pinjaman\_sebelumnya.

Berdasarkan hasil model algoritma *neural network* berbasis *particle swarm optimization* dapat dilakukan uji coba pada nilai *training cycles* ditentukan dengan cara memasukkan nilai *range* dari 100 sampai dengan 1500 untuk *training cycles*, serta nilai 0.3 untuk *learning rate* dan 0.2 untuk *momentum* dengan menggunakan data training. Berikut ini adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *training cycles*:

Tabel 4.13 Percobaan Penentuan Nilai *Training Cycles Neural Network* berbasis PSO

Training Cycles	Learning Rate	Momentum	NN		NN+PSO	
			Accuracy	AUC	Accuracy	AUC
100	0,3	0,2	89,55	0,938	91,62	0,942
200	0,3	0,2	89,7	0,938	92,41	0,945
300	0,3	0,2	90,97	0,942	92,56	0,939
400	0,3	0,2	90,97	0,941	93,36	0,945
500	0,3	0,2	92,23	0,941	93,99	0,946
600	0,3	0,2	92,71	0,942	93,99	0,946
700	0,3	0,2	92,77	0,945	95,57	0,956
800	0,3	0,2	93,03	0,945	95,25	0,956
900	0,3	0,2	93,18	0,943	94,93	0,951
1000	0,3	0,2	93,03	0,945	95,25	0,943
1100	0,3	0,2	93,18	0,945	95,41	0,955
1200	0,3	0,2	93,02	0,944	95,26	0,946
1300	0,3	0,2	93,65	0,943	95,73	0,972
1400	0,3	0,2	93,34	0,943	95,89	0,956
1500	0,3	0,2	93,65	0,945	95,89	0,961

Berdasarkan hasil uji coba pada nilai *training cycles* dipilih berdasarkan nilai akurasi dan nilai *Area Under Curve* (AUC) terbesar yang dihasilkan, dengan nilai 1500 pada *training Cycles*. Kemudian nilai tersebut akan digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Learning Rate*.

Nilai *Learning Rate* ditentukan dengan cara melakukan dengan uji coba memasukkan nilai dengan range 0.1 sampai dengan 0.9. Nilai *Training cycles* dipilih dari percobaan sebelumnya yaitu 1500, sedangkan 0.2 digunakan untuk nilai momentum. Berikut adalah hasil dari eksperimen yang telah dilakukan untuk menentukan nilai *Learning Rate*.

Tabel 4.14 Percobaan Penentuan nilai *Learning Rate* dengan *Neural Network* berbasis PSO

Learning Rate	Training Cycles	Momentum	NN		NN+PSO	
			Accuracy	AUC	Accuracy	AUC
0,1	1500	0,2	92,34	0,948	95,41	0,956
0,2	1500	0,2	92,65	0,947	96,20	0,957
0,3	1500	0,2	92,65	0,945	95,89	0,961
0,4	1500	0,2	93,29	0,949	95,58	0,938
0,5	1500	0,2	92,97	0,951	96,20	0,945
0,6	1500	0,2	92,5	0,943	96,04	0,95
0,7	1500	0,2	92,66	0,940	95,59	0,951
0,8	1500	0,2	93,44	0,957	95,41	0,955
0,9	1500	0,2	92,96	0,950	95,57	0,950

Berdasarkan hasil uji coba pada nilai *Learning Rate* dipilih berdasarkan nilai akurasi dan nilai *Area Under Curve* (AUC) terbesar yang dihasilkan, dengan nilai 0.5 pada *Learning Rate* dan 1500 pada *training Cycles*. Kemudian nilai tersebut akan digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Momentum*.

Nilai momentum ditentukan dengan cara melakukan eksperimen dengan cara memasukkan nilai training cycles 1500 dan *learning rate* 0.5, dan memasukkan nilai momentum dengan range 0.1 sampai dengan 0.9. Dengan nilai *training cycles* 1500 dan *learning rate* 0.5 berdasarkan percobaan sebelumnya. Berikut adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai momentum:

Tabel 4.15 Percobaan Penentuan nilai Momentum dengan  
*Neural Network* berbasis PSO

Momentum	Training Cycles	NN			NN + PSO		
		Learning Rate	Accuracy	AUC	Learning Rate	Accuracy	AUC
0,0	1500	0,8	92,34	0,930	0,5	95,26	0,935
0,1	1500	0,8	92,81	0,947	0,5	96,20	0,953
0,2	1500	0,8	93,44	0,957	0,5	96,20	0,945
0,3	1500	0,8	93,6	0,959	0,5	95,42	0,953
0,4	1500	0,8	92,02	0,926	0,5	95,89	0,958
0,5	1500	0,8	91,71	0,937	0,5	95,57	0,952
0,6	1500	0,8	91,7	0,928	0,5	96,20	0,943
0,7	1500	0,8	89,66	0,896	0,5	94,94	0,951
0,8	1500	0,8	88,22	0,772	0,5	94,47	0,941
0,9	1500	0,8	92,34	0,795	0,5	91,62	0,871

Berdasarkan hasil uji coba diatas, maka untuk parameter *neural network* dipilih nilai 1500 untuk nilai *training cycles*, 0.5 untuk nilai *learning rate* dan 0.2 untuk nilai *gfmomentum*.

Langkah berikutnya adalah menentukan jumlah *hidden layer* dan *neuron sizes* untuk mencari nilai akurasi dan AUC yang paling tinggi. Pada jumlah *hidden layer* sebanyak 1, dilakukan percobaan pada *neuron size* dengan *range* 1 sampai dengan sise 25.

Tabel 4.16 Percobaan Untuk Menentukan  
Neuron size NN-PSO

Hidden Layer	NN		NN +PSO	
	Accuracy	AUC	Accuracy	AUC
1	88,99	0,912	91,18	0,919
2	88,52	0,9	90,89	0,918
3	86,93	0,898	91,5	0,925
4	90,24	0,896	91,83	0,909
5	90,26	0,906	90,26	0,912
6	91,06	0,919	90,39	0,91
7	91,05	0,927	91,26	0,902
8	92,15	0,924	93,88	0,933
9	93,1	0,929	94,2	0,915
10	92	0,918	93,58	0,92
11	92,15	0,912	94,05	0,928
12	92,31	0,914	94,36	0,921
13	91,85	0,914	94,67	0,929
14	92,47	0,907	94,21	0,926
15	92,15	0,913	94,51	0,918
16	92,95	0,929	94,68	0,926
17	93,1	0,91	94,21	0,93
18	93,26	0,925	96,67	0,965
19	93,73	0,914	94,85	0,933
20	93,58	0,926	94,84	0,93
21	92,95	0,917	94,63	0,923
22	93,83	0,944	93,83	0,944
23	92,94	0,921	94,67	0,933
24	93,42	0,919	94,68	0,93
25	93,41	0,929	94,84	0,932

Berdasarkan tabel perbandingan tingkat kelancaran kredit antara model *Neural Network* dengan *Neural Network* berbasis PSO. Hasil terbaik pada percobaan diatas adalah dengan satu hidden layer dengan size 18 dengan *accuracy* yang dihasilkan sebesar 96,67% dan AUC-nya 0.965

#### 4.2.5 Model Algoritma Klasifikasi *Support Vector Machine*

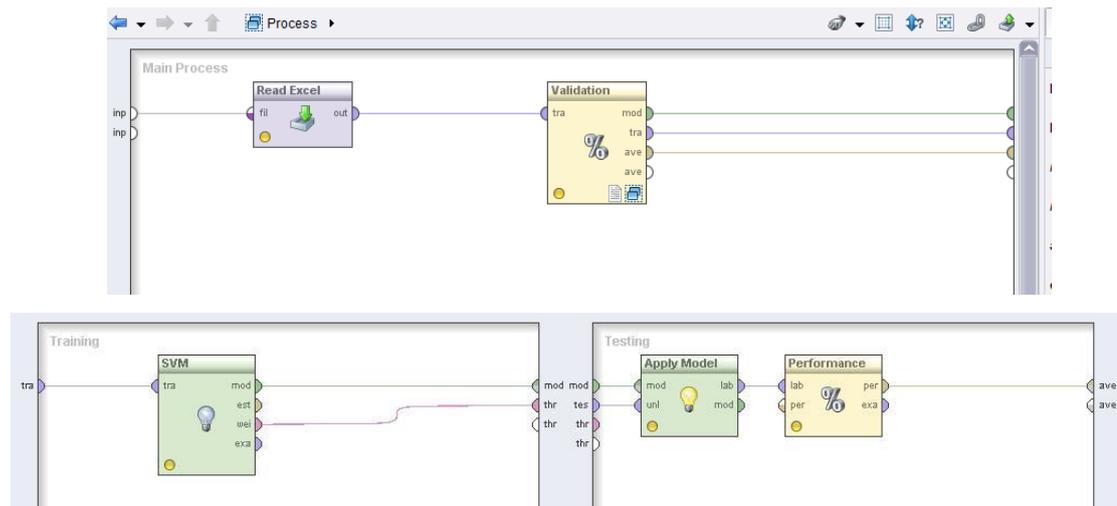
penelitian ini tipe atau parameter model SVM yang digunakan adalah Gamma, C dan Epsilon.

Sebelum itu dilakukan validasi 10 split dengan hasil sebagai berikut :

Kfold Cross Validation ( 10 Split SVM)									
Split 1	Split 2	Split 3	Split 4	Split 5	Split 6	Split 7	Split 8	Split 9	Split 10
88,6	89,62	88,6	89,63	89,05	89,04	88,84	90,05	89,04	89,02
Average = 89,15									

Dari hasil tersebut maka yang digunakan untuk proses pengujian adalah data dari split 5.

Fungsi *Kernel* digunakan untuk memapping data dalam pola non linier ke dalam pola yang linier dengan memindahkan data ke dimensi yang lebih tinggi, di mana hyperplane dapat memisahkan kelas data. Fungsi *kernel* yang biasa digunakan yaitu : *Linier*, *Radial Basis Function (RBF)*, *Polynomial* dan *Sigmoid*.



Gambar 4.9 Desain model algoritma *support vector machine*

Selanjutnya dilakukan eksperimen terhadap *gamma*, *C*, dan *epsilon* dari 3 kernel yang diuji yaitu *Radial*, *Dot*, dan *Polynomial*. Hasilnya adalah sebagai berikut:

Tabel 4.17 Hasil Eksperimen *Support Vector Machine*

No	Kernel Type	C	Epsilon	Gamma	Accuracy	AUC
1	Dot	0	0	-	88,93	0,942
2	Dot	1	0	-	89,57	0,946
3	Dot	2	0	-	89,25	0,942
4	Dot	4	0	-	89,41	0,946
5	Dot	8	0	-	89,25	0,942
6	Dot	16	0	-	88,62	0,941
7	Dot	32	0	-	87,03	0,92
8	Dot	64	0	-	87,51	0,939
9	Dot	128	0	-	87,95	0,894
10	Dot	256	0	-	89,85	0,932
11	Dot	512	0	-	83,82	0,772
12	Dot	1024	0	-	80,99	0,682
13	Dot	1024	1	-	84,79	0,5
14	Dot	1024	2	-	84,79	0,5
15	Dot	2048	2	-	84,79	0,5
16	Dot	4096	2	-	84,79	0,5
17	Dot	4096	4	-	84,79	0,5
18	Dot	4096	8	-	84,79	0,5
19	Dot	8192	16	-	84,79	0,5
20	Dot	8192	32	-	84,79	0,5
21	Dot	16384	16	-	84,79	0,5
22	Dot	16384	32	-	84,79	0,5
23	radial	0	0	1	84,94	0,729
24	radial	0	0	2	85,74	0,741
25	radial	0	0	4	86,21	0,751
26	radial	0	0	8	86,37	0,755
27	radial	0	0	16	86,52	0,739
28	radial	1	0	16	86,52	0,739
29	radial	2	0	16	86,37	0,739
30	radial	4	0	16	86,37	0,741
31	radial	8	0	16	86,37	0,741
32	radial	16	0	16	86,37	0,74
33	radial	32	0	16	86,37	0,742
34	radial	64	0	16	86,37	0,741
35	radial	128	0	16	86,37	0,741
36	radial	128	1	16	84,79	0,5
37	radial	128	2	16	84,79	0,5
38	radial	256	0	16	86,37	0,74
39	radial	512	0	16	86,37	0,74

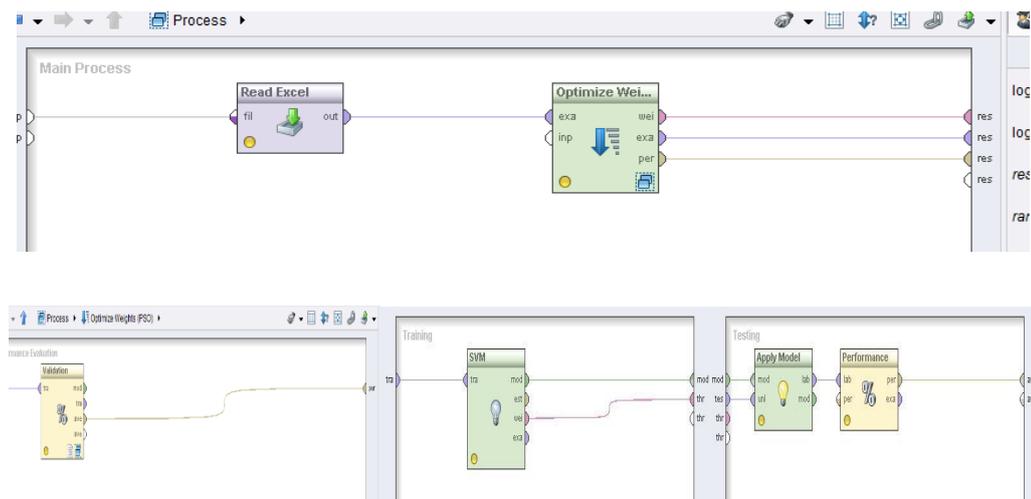
Tabel 4.17 Hasil Eksperimen *Support Vector Machine*

No	Kernel Type	C	Epsilon	Gamma	Accuracy	AUC
40	Polynomial	0	0	-	87,64	0,734
41	Polynomial	2	0	-	80,95	0,769
42	Polynomial	4	0	-	77,8	0,726
43	Polynomial	8	0	-	79,38	0,773
44	Polynomial	16	0	-	85,1	0,79
45	Polynomial	32	0	-	86,06	0,869
46	Polynomial	64	0	-	85,28	0,845
47	Polynomial	64	1	-	84,79	0,5
48	Polynomial	64	2	-	84,79	0,5
49	Polynomial	128	4	-	84,79	0,5
50	Polynomial	256	8	-	84,79	0,5
51	Polynomial	512	16	-	84,79	0,5
52	Polynomial	1024	32	-	84,79	0,5

Berdasarkan tabel perbandingan tingkat kelancaran kredit antara model *Support Vector Machine*. Hasil terbaik pada percobaan diatas adalah dengan kernal type Dot dengan *accuracy* yang dihasilkan sebesar 89.85% dan AUC-nya 0.932

#### 4.2.6 Model Algoritma Klasifikasi *Support Vector Machine* dengan *Particle Swarm Optimazation*

Berikut adalah gambar pengujian algoritma *Support Vector Machine* berbasis *Particle Swarm Optimization* menggunakan metode *K-Fold Cross Validation* dengan menggunakan *RapidMiner*

Gambar 4.10 Desain model algoritma *support vector machine* berbasis PSO

Berdasarkan hasil dari uji coba dengan menggunakan algoritma *Support Vector Machine* maka data *training* tersebut akan diseleksi dengan attribute yang digunakan yaitu Akun\_bank, lama\_pinjam, jamina, jumlah\_kredit, jumlah\_angsuran, umur, pekerjaan, masa\_kerja, gaji, persentase\_pemotongan\_gaji, jenis\_kelamin, status, pinjaman\_sebelumnya dan 1 atribut sebagai label yaitu *response*.

Tabel 4.18 Bobot Atribut dengan Metode SVM dengan PSO

Attribute	Weight
Akun_bank	0
Lama_pinjam	1
jaminan	0
jumlah_kredit	1
jumlah_angsuran	0
umur	0
pekerjaan	0
masa_kerja	1
gaji	0
persentase_pemotongan_gaji	0.528
jenis_kelamin	1
status	1
pinjaman_sebelumnya	0

Dari hasil uji coba menggunakan algoritma *support vector machine* berbasis *particle swarm optimization* diperoleh beberapa atribut-atribut yang berpengaruh terhadap bobot atribut yaitu: lama\_pinjam, jumlah\_kredit, masa\_kerja, persentase\_pemotongan\_gaji, jenis\_kelamin, status. Berdasarkan hasil uji coba akun\_bank, jaminan, jumlah\_angsuran, pekerjaan, umur, gaji, pinjaman\_sebelumnya tidak berpengaruh terhadap bobot attribute

Selanjutnya dilakukan eksperimen terhadap *gamma*, *C*, dan *epsilon* dari 3 kernel yang diuji yaitu *Radial*, *Dot*, dan *Polynomial*. Hasilnya adalah sebagai berikut:

Tabel 4.19 Hasil Eksperimen *Support Vector Machine* berbasis *Particle Swarm Optimazation*

No	Kernel Type	C	Epsilon	Gamma	Accuracy	AUC
1	Dot	0	0	-	89,55	0,934
2	Dot	1	0	-	90,66	0,935
3	Dot	2	0	-	90,5	0,941
4	Dot	4	0	-	90,49	0,946
5	Dot	8	0	-	90,5	0,941
6	Dot	16	0	-	90,48	0,941
7	Dot	32	0	-	90,65	0,939
8	Dot	64	0	-	90,66	0,933
9	Dot	128	0	-	90,49	0,943
10	Dot	256	0	-	90,65	0,946
11	Dot	512	0	-	90,48	0,941
12	Dot	1024	0	-	84,95	0,861
13	Dot	1024	1	-	84,79	0,5
14	Dot	1024	2	-	84,79	0,5
15	Dot	2048	2	-	84,79	0,5
16	Dot	4096	2	-	84,79	0,5
17	Dot	4096	4	-	84,79	0,5
18	Dot	4096	8	-	84,79	0,5
19	Dot	8192	16	-	84,79	0,5
20	Dot	8192	32	-	84,79	0,5
21	Dot	16384	16	-	84,79	0,5
22	Dot	16384	32	-	84,79	0,5
23	radial	0	0	1	91,29	0,919
24	radial	0	0	2	92,07	0,951
25	radial	0	0	4	92,56	0,965
26	radial	0	0	8	94,14	0,971
27	radial	0	0	16	94,6	0,976
28	radial	1	0	16	94,6	0,976
29	radial	2	0	16	95,57	0,987
30	radial	4	0	16	96,19	0,986
31	radial	8	0	16	95,72	0,988
32	radial	16	0	16	95,88	0,993
33	radial	32	0	16	96,19	0,988
34	radial	64	0	16	95,88	0,994
35	radial	128	0	16	96,04	0,965
36	radial	128	1	16	84,79	0,5
37	radial	128	2	16	84,79	0,5
38	radial	256	0	16	96,04	0,969
39	radial	512	0	16	96,2	0,989

Tabel 4.19 Hasil Eksperimen *Support Vector Machine* berbasis *Particle Swarm Optimazation*(Lanjutan)

No	Kernel Type	C	Epsilon	Gamma	Accuracy	AUC
40	Polynominal	0	0	-	89,07	0,744
41	Polynominal	2	0	-	84,79	0,5
42	Polynominal	4	0	-	90,49	0,886
43	Polynominal	8	0	-	90,33	0,886
44	Polynominal	16	0	-	91,59	0,897
45	Polynominal	32	0	-	92,08	0,913
46	Polynominal	64	0	-	90,49	0,874
47	Polynominal	64	1	-	84,79	0,5
48	Polynominal	64	2	-	84,79	0,5
49	Polynominal	128	4	-	84,79	0,5
50	Polynominal	256	8	-	84,79	0,5
51	Polynominal	512	16	-	84,79	0,5
52	Polynominal	1024	32	-	84,79	0,5

Berdasarkan tabel perbandingan tingkat kelancaran kredit antara model *Support Vector Machine* berbasis *Particle Swarm Optimazation*. Hasil terbaik pada percobaan diatas adalah dengan kernal type Radial dengan *accuracy* yang dihasilkan sebesar 96.2% dan AUC-nya 0.989

### 4.3 Hasil Pengujian Algoritma Klasifikasi Data Mining

Evaluasi model dengan confusion matrix akan membentuk matrix yang terdiri dari True Positif atau tupel positif dan True Negatif atau tupel negatif

#### 4.3.1 Evaluasi model C4.5 dengan Confusion Matrix

Confusion matrix membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada gambar IV.10, akurasi C4.5 sebesar 93.81 % . dimana Jumlah True Positive (TP) adalah 525 , Untuk False Negative (FN) adalah 29 , untuk False Positive (FP) adalah 10 dan Untuk True Negative (TN) adalah 67

accuracy: 93.81% +/- 1.96% (mikro: 93.82%)			
	true Lancar	true Bermasalah	class precision
pred. Lancar	525	29	94.77%
pred. Bermasalah	10	67	87.01%
class recall	98.13%	69.79%	

Gambar 4.11 Hasil Akurasi Algoritmat C4.5

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*,

*ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$\begin{aligned}
 Acc &= \frac{tp+tn}{tp+tn+fp+fn} & acc &= \frac{525+67}{525+67+10+29} \\
 Sensitivity &= \frac{tp}{tp+fn} & Sensitivity &= \frac{525}{525+29} \\
 Specificity &= \frac{tn}{tn+fp} & Specificity &= \frac{67}{67+10} \\
 PPV &= \frac{tp}{tp+fp} & PPV &= \frac{525}{525+10} \\
 NPV &= \frac{tn}{tn+fn} & NPV &= \frac{67}{67+29}
 \end{aligned}$$

Tabel 4.20 Nilai *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv* C4.5

	Nilai ( % )
<i>Accuracy</i>	93,81
<i>Sensitivity</i>	94,76
<i>Specificity</i>	87,01
<i>PPV</i>	98,13
<i>NPV</i>	69,79

#### 4.3.2 Evaluasi model C4.5 berbasis PSO dengan Confusion Matrix

Confusion matrix membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada gambar IV.11 , akurasi C4.5 berbasis PSO sebesar 94.29 % . dimana Jumlah True Positive (TP) adalah 529 , Untuk False Negative (FN) adalah 30 , untuk False Positive (FP) adalah 6 dan Untuk True Negative (TN) adalah 66

accuracy: 94.29% +/- 2.50% (mikro: 94.29%)			
	true Lancar	true Bermasalah	class precision
pred. Lancar	529	30	94.63%
pred. Bermasalah	6	66	91.67%
class recall	98.88%	68.75%	

Gambar 4.12 Hasil Akurasi Algorimat C4.5 berbasis PSO

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*,

*ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$\begin{aligned}
 Acc &= \frac{tp+tn}{tp+tn+fp+fn} & acc &= \frac{529+66}{529+66+6+30} \\
 Sensitivity &= \frac{tp}{tp+fn} & Sensitivity &= \frac{529}{529+30} \\
 Specificity &= \frac{tn}{tn+fp} & Specificity &= \frac{66}{66+6} \\
 PPV &= \frac{tp}{tp+fp} & PPV &= \frac{529}{529+6} \\
 NPV &= \frac{tn}{tn+fn} & NPV &= \frac{66}{66+30}
 \end{aligned}$$

Tabel 4.21 Nilai *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv*  
C4.5 berbasis PSO

	Nilai ( % )
<i>Accuracy</i>	94.29
<i>Sensitivity</i>	94,63
<i>Specificity</i>	91,66
<i>PPV</i>	98,87
<i>NPV</i>	68,75

### 4.3.3 Evaluasi model *Support Vector Machine* dengan *Confusion Matrix*

Confusion matrix membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada gambar IV.12 , akurasi *Support Vector Machine* sebesar 89,85 % . dimana Jumlah True Positive (TP) adalah 524 , Untuk False Negative (FN) adalah 53 , untuk False Positive (FP) adalah 11 dan Untuk True Negative (TN) adalah 43

	true Lancar	true Bermasalah	class precision
pred. Lancar	524	53	90.81%
pred. Bermasalah	11	43	79.63%
class recall	97.94%	44.79%	

Gambar 4.13 Hasil Akurasi Algoritma *Support Vector Machine*

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*,

*ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$\begin{aligned}
 Acc &= \frac{tp+tn}{tp+tn+fp+fn} & acc &= \frac{524+43}{524+43+11+53}
 \end{aligned}$$

$$\text{Sensitivity} = \frac{tp}{tp+fn}$$

$$\text{Sensitivity} = \frac{524}{524+53}$$

$$\text{Specificity} = \frac{tn}{tn+fp}$$

$$\text{Specificity} = \frac{43}{43+11}$$

$$\text{PPV} = \frac{tp}{tp+fp}$$

$$\text{PPV} = \frac{524}{524+11}$$

$$\text{NPV} = \frac{tn}{tn+fn}$$

$$\text{NPV} = \frac{43}{43+53}$$

Tabel 4.22 Nilai Accuracy, Sensitivity, Specificity, ppv, dan npv

### Support Vector Machine

	Nilai ( % )
Accuracy	89,85
Sensitivity	90,81
Specificity	79,62
PPV	97,94
NPV	44,79

#### 4.3.4 Evaluasi model Support Vector Machine berbasis PSO dengan Confusion Matrix

Confusion matrix membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada gambar IV.13 , akurasi Support Vector Machine berbasis PSO sebesar 96.20 % . dimana Jumlah True Positive (TP) adalah 531 , Untuk False Negative (FN) adalah 20 , untuk False Positive (FP) adalah 4 dan Untuk True Negative (TN) adalah 76

accuracy: 96.20% +/- 1.91% (mikro: 96.20%)			
	true Lancar	true Bermasalah	class precision
pred. Lancar	531	20	96.37%
pred. Bermasalah	4	76	95.00%
class recall	99.25%	79.17%	

Gambar 4.14 Hasil Akurasi Algoritma Support Vector Machine berbasis PSO

Dari hasil diatas dapat di hitung untuk mencari nilai accuracy, sensitivity, specificity,

ppv, dan npv hasilnya pada persamaan dibawah ini:

$$\text{Acc} = \frac{tp+tn}{tp+tn+fp+fn}$$

$$\text{acc} = \frac{524+43}{524+43+11+53}$$

$$\text{Sensitivity} = \frac{tp}{tp+fn}$$

$$\text{Sensitivity} = \frac{524}{524+53}$$

$$Specificity = \frac{tn}{tn+fp}$$

$$Specificity = \frac{43}{43+11}$$

$$PPV = \frac{tp}{tp+fp}$$

$$PPV = \frac{524}{524+11}$$

$$NPV = \frac{tn}{tn+fn}$$

$$NPV = \frac{43}{43+53}$$

Tabel 4.23 Nilai *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv*  
*Support Vector Machine* berbasis PSO

	Nilai ( % )
<i>Accuracy</i>	89,85
<i>Sensitivity</i>	90,81
<i>Specificity</i>	79,62
<i>PPV</i>	97,94
<i>NPV</i>	44,79

#### 4.3.5 Evaluasi model *Neural Network* dengan *Confusion Matrix*

*Confusion matrix* membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada gambar IV.14 , akurasi *Neural Network* 93,83 % . dimana Jumlah True Positive (TP) adalah 517 , Untuk False Negative (FN) adalah 21 , untuk False Positive (FP) adalah 18 dan Untuk True Negative (TN) adalah 76

accuracy: 93.83% +/- 2.68% (mikro: 93.83%)			
	true Lancar	true Bermasalah	class precision
pred. Lancar	517	21	96.10%
pred. Bermasalah	18	76	80.85%
class recall	96.64%	78.35%	

Gambar 4.15 Hasil Akurasi Algoritma *Neural Network*

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*,

*ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$Acc = \frac{tp+tn}{tp+tn+fp+fn}$$

$$acc = \frac{517+76}{517+76+18+21}$$

$$Sensitivity = \frac{tp}{tp+fn}$$

$$Sensitivity = \frac{517}{517+21}$$

$$Specificity = \frac{tn}{tn+fp}$$

$$Specificity = \frac{76}{76+18}$$

$$PPV = \frac{tp}{tp+fp}$$

$$NPV = \frac{tn}{tn+fn}$$

$$PPV = \frac{517}{517+18}$$

$$NPV = \frac{76}{76+21}$$

Tabel 4.24 Nilai *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv* *Neural Network*

	Nilai ( % )
<i>Accuracy</i>	93,83
<i>Sensitivity</i>	96,09
<i>Specificity</i>	80,85
<i>PPV</i>	96,63
<i>NPV</i>	78,35

#### 4.3.6 Evaluasi model *Neural Network* berbasis PSO dengan Confusion Matrix

Confusion matrix membentuk matriks yang terdiri dari true positif atau tupel positif dan true negatif atau tupel negatif. Pada tabel IV.15 , akurasi *Neural Network* berbasis PSO 96,67 % . dimana Jumlah True Positive (TP) adalah 526 , Untuk False Negative (FN) adalah 12 , untuk False Positive (FP) adalah 9 dan Untuk True Negative (TN) adalah 85

	true Lancar	true Bermasalah	class precision
pred. Lancar	526	12	97.77%
pred. Bermasalah	9	85	90.43%
class recall	98.32%	87.63%	

Gambar 4.16 Hasil Akurasi Algoritma *Neural Network* berbasis PSO

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*,

*ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$Acc = \frac{tp+tn}{tp+tn+fp+fn}$$

$$Sensitivity = \frac{tp}{tp+fn}$$

$$Specificity = \frac{tn}{tn+fp}$$

$$PPV = \frac{tp}{tp+fp}$$

$$acc = \frac{526+85}{526+85+9+12}$$

$$Sensitivity = \frac{526}{526+12}$$

$$Specificity = \frac{85}{85+9}$$

$$PPV = \frac{526}{526+9}$$

$$NPV = \frac{tn}{tn+fn}$$

$$NPV = \frac{85}{85+12}$$

Tabel 4.25 Nilai *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv* Neural Network berbasis PSO

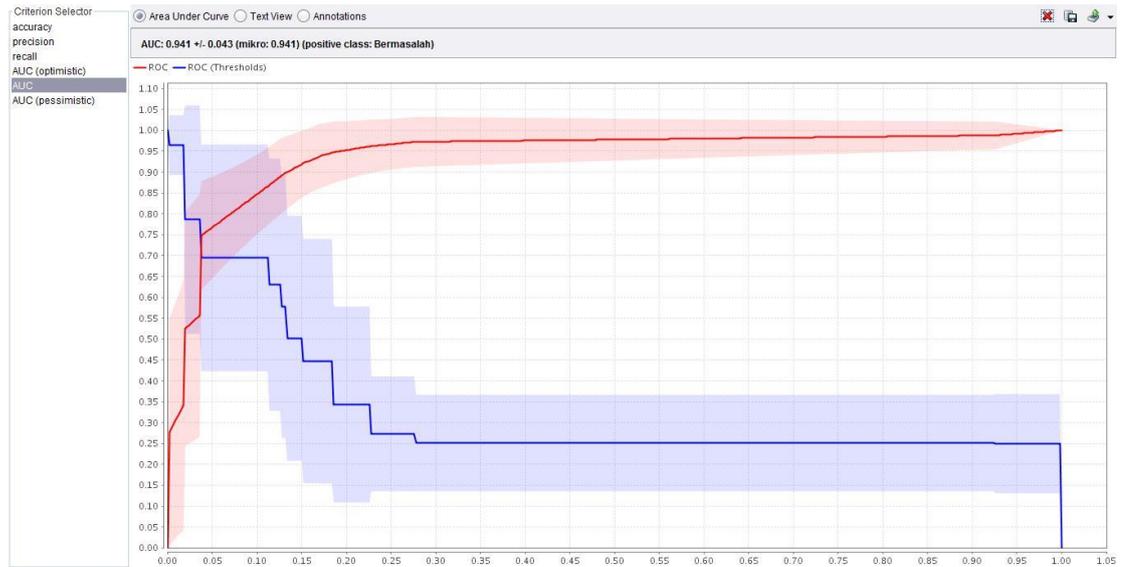
	Nilai ( % )
<i>Accuracy</i>	96,67
<i>Sensitivity</i>	97,76
<i>Specificity</i>	90,42
<i>PPV</i>	98,31
<i>NPV</i>	87,62

#### 4.3.7 Evaluasi Dengan ROC Curve

Akurasi Memiliki tingkat Diagnosa yaitu (Gorunescu, 2011)

1. Akurasi bernilai 0.90 – 1.00 = *Excellent Classification*
2. Akurasi bernilai 0.80 – 0.90 = *Good Classification*
3. Akurasi bernilai 0.70 – 0.80 = *Fair Classification*
4. Akurasi bernilai 0.60 – 0.70 = *Poor Classification*
5. Akurasi bernilai 0.50 – 0.60 = *Failure*

Pada Gambar 4.17 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma C4.5 sebesar 0.941 dengan tingkat diagnosa *Excellent Classification*



Gambar 4.17 Nilai AUC C4.5 dalam *ROC Curve*

Pada Gambar 4.18 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma C4.5 berbasis PSO sebesar 0.936 dengan tingkat diagnosa *Excellent Classification*



Gambar 4.18 Nilai AUC C4.5 berbasis PSO dalam *ROC Curve*

Pada Gambar 4.19 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma *Support Vector Machine* sebesar 0.932 dengan tingkat diagnosa *Excellent Classification*



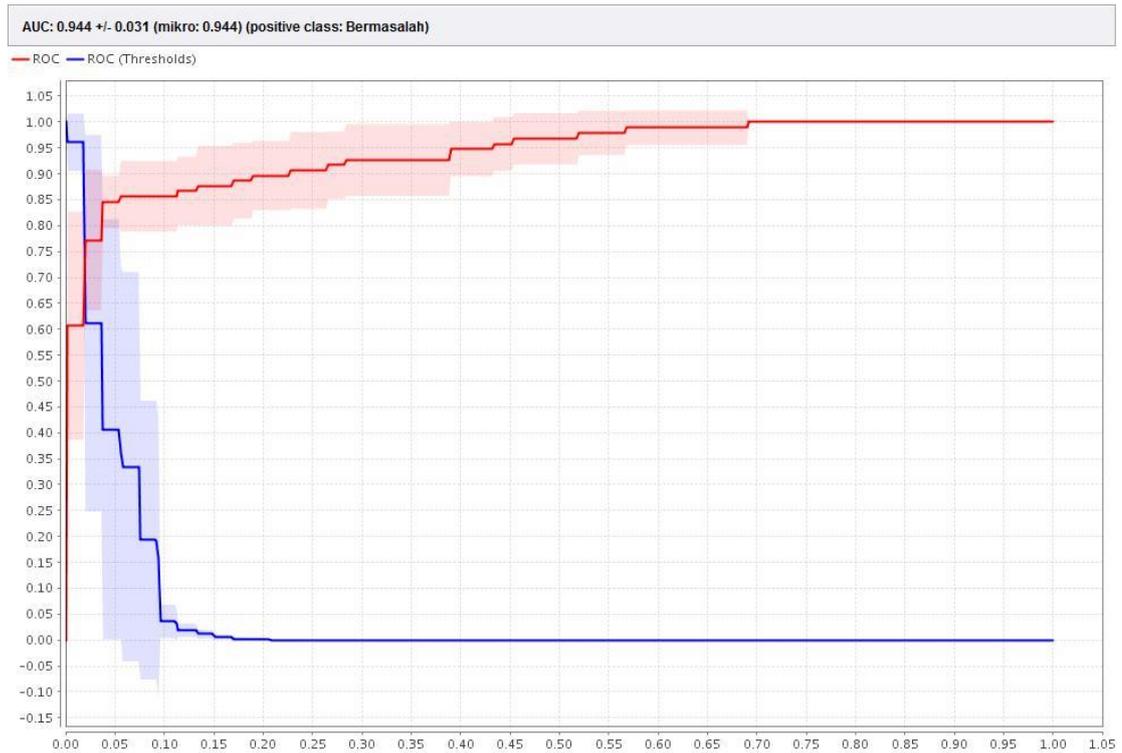
Gambar 4.19 Nilai AUC *Support Vector Machine* dalam *ROC Curve*

Pada Gambar 4.20 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma *Support Vector Machine* berbasis PSO sebesar 0.989 dengan tingkat diagnosa *Excellent Classification*



Gambar 4.20 Nilai AUC *Support Vector Machine* berbasis PSO dalam *ROC Curve*

Pada Gambar 4.21 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma *Neural Network* sebesar 0.944 dengan tingkat diagnosa *Excellent Classification*



Gambar 4.21 Nilai AUC *Neural Network*

Pada Gambar 4.22 menunjukkan grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma *Neural Network* Berbasis PSO sebesar 0.965 dengan tingkat diagnosa *Excellent Classification*



Gambar 4.22 Nilai AUC *Neural Network* berbasis PSO

Berdasarkan dari analisa pengujian masing-masing algoritma di atas maka dapat dirangkumkan hasilnya pada tabel 4.23 sebagai berikut

Tabel 4.26 Perbandingan *Performance* Metode

	<b>C4.5</b>	<b>C4.5 + PSO</b>	<b>SVM</b>	<b>SVM + PSO</b>	<b>NN</b>	<b>NN + PSO</b>
<b>Accuracy</b>	93,81%	94,29%	89,85%	96,20%	93,83%	96,67%
<b>AUC</b>	0,941	0,936	0,932	0,989	0,944	0,965

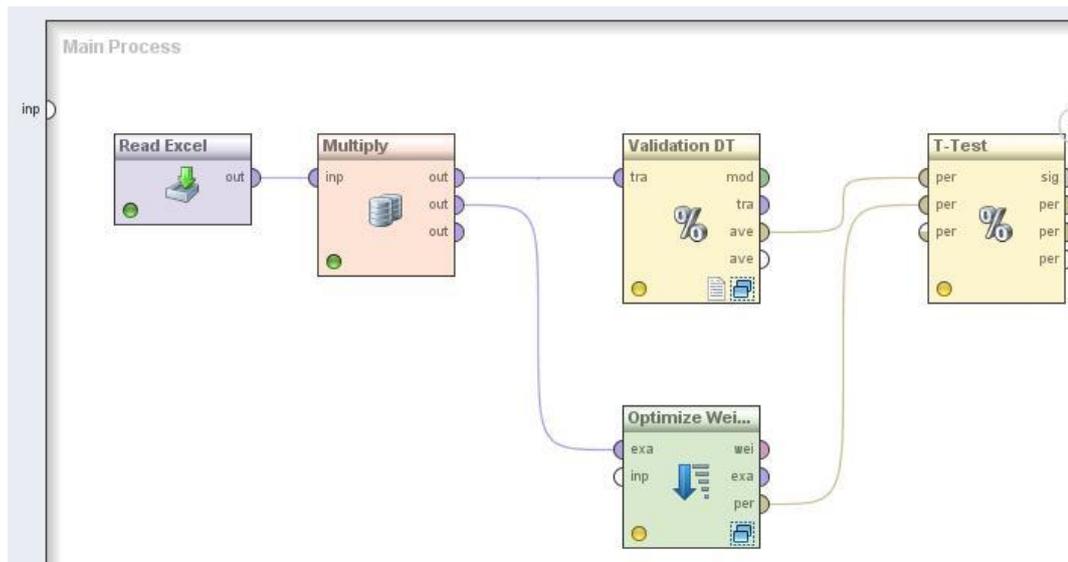
Melihat dari kedua hasil pengujian di atas (*Accuracy* dan *AUC*) ternyata masih belum bisa ditentukan algoritma apa yang terbaik. Menurut pengujian berdasarkan *Accuracy*, algoritma terbaik adalah *Neural Network* berbasis *Particle Swarm Optimazation* sedangkan menurut pengujian berdasarkan *ROC Curve* (*AUC*), algoritma yang terbaik adalah *Algoritma Support Vector Machine* berbasis *Particle Swarm Optimazation*. Untuk penentuan lebih lanjut akan digunakan pengujian dengan memanfaatkan uji statistik yaitu dengan menggunakan uji T-Test (Santosa, 2010).

Pada pengujian ini akan dibandingkan antar dua algoritma secara bergantian sehingga dihasilkan perbandingan keseluruhan seperti pada tabel berikut ini :

Tabel 4.27 Uji Statistik T-Test

	<b>C4.5</b>	<b>C4.5 + PSO</b>	<b>SVM</b>	<b>SVM + PSO</b>	<b>NN</b>	<b>NN + PSO</b>
<b>C4.5</b>		0,116	<b>0,044</b>	<b>0,02</b>	0,449	<b>0,002</b>
<b>C4.5 + PSO</b>	<b>0,036</b>		<b>0,008</b>	0,100	0,912	<b>0,044</b>
<b>SVM</b>	<b>0,016</b>	<b>0,000</b>		<b>0,000</b>	<b>0,000</b>	<b>0,000</b>
<b>SVM + PSO</b>	<b>0,001</b>	0,076	<b>0,001</b>		<b>0,043</b>	0,765
<b>NN</b>	<b>0,006</b>	0,525	<b>0,003</b>	0,340		0,105
<b>NN + PSO</b>	<b>0,001</b>	0,133	<b>0,001</b>	0,704	<b>0,025</b>	

Hasil tabel di atas didapatkan dari pengujian T-Test dengan membandingkan dua algoritma secara bergantian seperti terlihat pada salah satu model di bawah ini :



Gambar 4.23 Model T-Test pada *Decision Tree* dan *Neural Network PSO*

Jika model di atas dijalankan maka akan mendapatkan hasil perhitungan statistik seperti pada gambar 4.24 di bawah ini :

### T-Test Significance

	0.922 +/- 0.031	0.967 +/- 0.025
0.922 +/- 0.031		<b>0.002</b>
0.967 +/- 0.025		

Probabilities for random values with the same result.

Bold values are smaller than  $\alpha=0.050$  which indicates a probably significant difference between the actual mean values!

Gambar 4.24 Hasil Model T-Test pada *Decision Tree* dan *Neural Network PSO*

Dari gambar 4.24 di atas dapat dianalisis bahwa algoritma *Neural Network* berbasis *Particle Swarm Optimazation* memiliki perbedaan nilai yang sangat dominan atau signifikan karena memiliki nilai probabilitas  $< 0,05$  yaitu :

- 0,002 terhadap algoritma C4.5
- 0,044 terhadap algoritma C4.5 berbasis PSO
- 0,000 terhadap algoritma *Support Vector Machine*

Sehingga dapat disimpulkan bahwa algoritma yang paling akurat untuk penentuan kelayakan pemberian kredit adalah *Neural Network* berbasis *Particle Swarm Optimazation* karena memiliki nilai uji *accuracy* tertinggi yaitu 96,67% dan

nilai uji statistik T-Test paling dominan terhadap algoritma lainnya walaupun memiliki nilai ujia AUC tidak paling tinggi yaitu 0,965.

Tabel 4.28 Hasil Perbandingan seluruh pengujian

	<b>C4.5</b>	<b>C4.5 + PSO</b>	<b>SVM</b>	<b>SVM + PSO</b>	<b>NN</b>	<b>NN + PSO</b>
<b>Accuracy</b>	93,81%	94,29%	89,85%	96,20%	93,83%	96,67%
<b>AUC</b>	0,941	0,936	0,932	0,989	0,944	0,965
<b>T-Test</b>	Dominan	Tidak Dominan	Dominan	Tidak Dominan	Dominan	Dominan

Melihat hasil perbandingan dari seluruh pengujian yang telah dilakukan seperti terangkum pada tabel 4.24 didapatkan 4 algoritma yang dominan yaitu C4.5, SVM, *Neural Network*, dan *Neural Network* berbasis *Particle Swarm Optimazation* dengan nilai akurasi untuk C4.5 *Accuracy* 93,81 % dan AUC 0,941. Algoritma *Support Vector Machine* dengan nilai akurasi 89,85% dan AUC 0,932. Algoritma *Neural Network* dengan nilai akurasi 93,83 % dan nilai AUC 0,944 serta Algoritma *Neural Network* berbasis *Particle Swarm Optimazation* dengan nilai akurasi 96,67 % dan nilai AUC 0,965. Dilihat dari dari hasil Akurasi dan nilai AUC ke empat algoritma yang dominan, maka dipilihlah algoritma *Neural Network* berbasis *Particle Swarm Optimazation* sebagai algoritma yang paling akurat karena memiliki nilai akurasi paling tinggi serta dominan diantara algoritma yang lain.

#### 4.4 Perancangan dan Pengujian Prototype

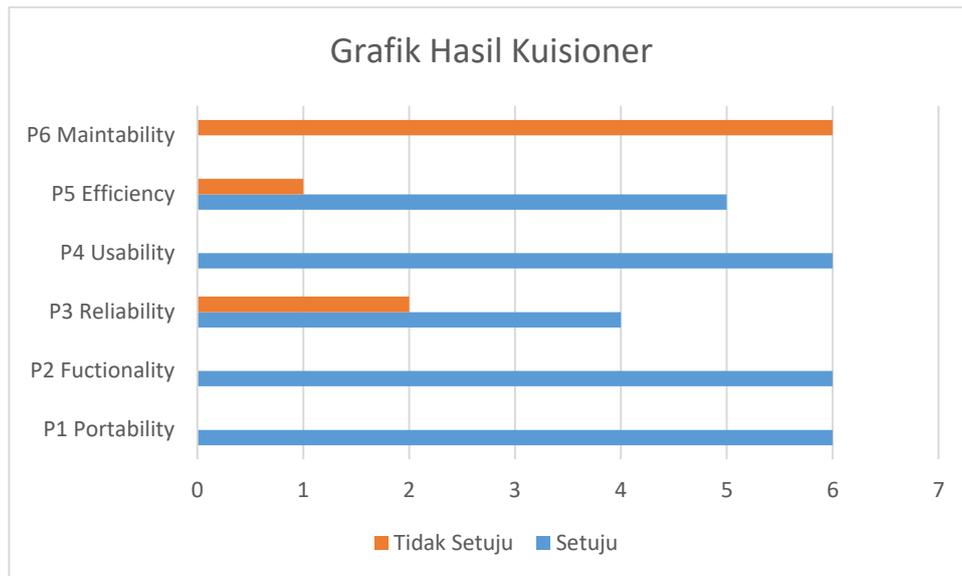
Dalam implementasi penelitian ini menggunakan aplikasi penentuan kelayakan kredit menggunakan Program microsoft Visual Basic 6.0 sebagai prototype . nama project adalah kredit.vbp dengan 1 buah form, Form Kredit. Pada form marshall berfungsi untuk menghasilkan nilai keputusan kelayakan pemberian kredit dari hasil 13 atribut inputan. Prototypenya sebagai berikut :

NO	Akun Bank	Lama Pinjam	Jaminan	Jumlah Kredit	Jumlah Angsuran	Umur	Pekerjaan	Jasa Kerja	Gaji	persentase Pemotongan	Jenis Kelamin	Status
1	Memiliki	96	Surat_beharga	108000000	2205000	45	PNS	4	5095000	43,28	Pria	Menikah
2	Memiliki	77	Surat_beharga	129000000	2911575	53	Guru	4	8814750	42,72	Pria	Menikah
3	Memiliki	36	Surat_beharga	80000000	2888889	50	Guru	4	8524480	41,72	Wanita	Menikah
4	Memiliki	48	Surat_beharga	100000000	3000000	47	Guru	4	7288953	41,16	Wanita	Menikah
5	Memiliki	84	Surat_beharga	110000000	2409524	41	Guru	4	6713616	35,89	Wanita	Menikah
6	Memiliki	87	Surat_beharga	190000000	4044325	53	Guru	4	8271147	48,8	Wanita	Menikah
7	Memiliki	84	Surat_beharga	112000000	2453334	45	Guru	4	6395016	36,42	Wanita	Menikah
8	Memiliki	96	Surat_beharga	108000000	2205000	45	PNS	4	6615600	33,33	Pria	Menikah
9	Memiliki	36	Surat_beharga	100000000	3563445	50	Guru	4	9048300	39,45	Pria	Menikah
10	Memiliki	74	Surat_beharga	179000000	3643871	53	PNS	21	9887021	42,11	Pria	Menikah

Gambar 4.25 Prototype Aplikasi Penentuan Kelayakan Kredit

Prototype yang peneliti buat menggunakan logika dari algoritma *Neural Network* berbasis *Particle Swarm Optimazation*. Karena dari hasil pengujian algoritma *Neural Network* berbasis *Particle Swarm Optimazation* terbukti lebih akurat dibandingkan dengan algoritma *Decision Tree* dan *Support Vector Machine*.

Setelah prototype dirancang dan diimplementasikan, diperlukan suatu pengujian kualitas perangkat lunak berdasarkan ISO 9126. Melalui penyebaran kuisioner yang dibagikan terhadap beberapa pemegang kepentingan seperti Analis, Administrasi, Bagian Resiko Perkreditan di lingkungan Bank DKI Jakarta Cab. Bendungan Hilir. 6 koresponden diambil sampelnya untuk menilai 6 karakter standar kualitas software pada prototype yang dirancang. Tergambar dalam kurva dibawah ini, sebagian besar koresponden menyatakan bahwa karakter *portability*, *functionality* dan *usability* sangat terpenuhi pada prototype ini. Sedangkan *efficiency* lebih dari cukup untuk prototype dimana 5 koresponden menyatakan sudah sesuai dan 1 koresponden menyatakan belum sesuai. Sedangkan *Reliability* lebih dari cukup untuk prototype dimana 4 koresponden menyatakan sudah sesuai dan 2 koresponden menyatakan belum sesuai. Sedangkan faktor *Maintability* masih belum terpenuhi dikarenakan masih perlu pengembangan lebih lanjut. Berikut gambar grafik dari hasil kuisioner ISO 9126 dari 6 koresponden yang terlibat.



Gambar 4.26 Grafik hasil Kuisisioner

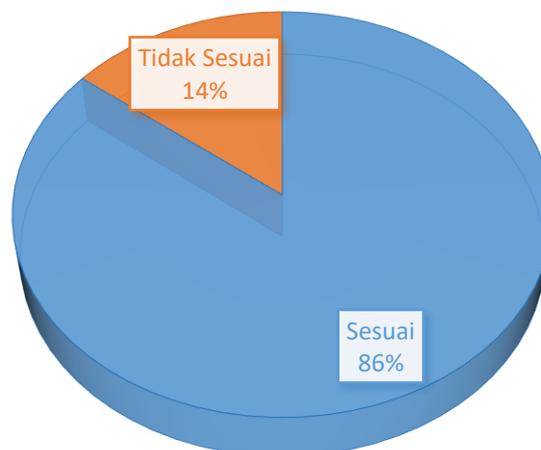
Selain pengujian ISO 9126 dilakukan juga pengujian *prototype* terhadap 10 % data *testing*. Proses ini dilakukan untuk mengetahui tingkat akurasi dari *prototype* yang telah di rancang menggunakan algoritma *Neural Network* berbasis *Particle Swarm Optimazation*. Berikut merupakan data testing yang akan di uji coba pada *prototype* aplikasi penentuan kelayakan kredit.

Tabel 4.29 Data Testing

akun_bank	lama_pinjam	jaminan	jumlah_kredit	jumlah_angsuran	umur	pekerjaan	masa_kerja	gaji	persentase_pemotongan_gaji	jenis_kelamin	status	pinjaman_sebelumnya	response
Memiliki	18	Surat_berharga	5000000	3277778	32	Karyawan_Swasta	2	7702500	42,55	Pria	Menikah	lunas	Lancar
Memiliki	11	Surat_berharga	23000000	2320910	55	Karyawan_Swasta	4	2500012	90,66	Pria	Menikah	lunas	Bermasalah
Memiliki	11	Surat_berharga	12000000	1210910	50	Karyawan_Swasta	4	2500012	47,3	Pria	Menikah	lunas	Lancar
Tidak_Memiliki	36	Surat_berharga	40000000	1511112	35	Guru	4	3251586	46,47	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
Tidak_Memiliki	24	Surat_berharga	25000000	1291667	30	Guru	4	2301980	39,72	Pria	Menikah	Tidak_ada_pinjaman	Lancar
Tidak_Memiliki	36	Surat_berharga	25000000	1291667	32	Karyawan_Swasta	3	3393819	27,83	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
Tidak_Memiliki	36	Surat_berharga	25000000	944445	44	Guru	4	4021716	23,48	Pria	Menikah	Tidak_ada_pinjaman	Lancar
Tidak_Memiliki	24	Surat_berharga	25000000	1291667	44	Karyawan_Swasta	4	4546089	28,41	Pria	Menikah	Tidak_ada_pinjaman	Lancar
Memiliki	24	Surat_berharga	68000000	3523219	40	Karyawan_Swasta	4	4346089	77,5	Pria	Menikah	Tidak_ada_pinjaman	Bermasalah
Tidak_Memiliki	60	Surat_berharga	25000000	944445	50	Guru	4	4900595	19,27	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
Memiliki	36	Surat_berharga	73000000	2777004	50	Guru	4	4900595	56,67	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
Tidak_Memiliki	24	Surat_berharga	29000000	1498334	41	Karyawan_Swasta	4	2948699	50,81	Wanita	Menikah	Tidak_ada_pinjaman	Bermasalah
Tidak_Memiliki	24	Surat_berharga	73000000	3808737	41	Karyawan_Swasta	4	2984700	129,17	Wanita	Menikah	lunas	Bermasalah
Memiliki	24	Surat_berharga	28000000	1446667	41	Karyawan_Swasta	4	2948699	49,06	Wanita	Menikah	Tidak_ada_pinjaman	Lancar
Memiliki	12	Surat_berharga	20000000	1833334	24	PNS	2	3689571	49,69	Wanita	Menikah	lunas	Bermasalah
Memiliki	12	Surat_berharga	19900000	1824167	24	PNS	2	3689572	49,44	Wanita	Menikah	lunas	Lancar
Tidak_Memiliki	36	Surat_berharga	20000000	747223	42	Karyawan_Swasta	3	4546912	16,43	Pria	Menikah	Tidak_ada_pinjaman	Lancar
Tidak_Memiliki	36	Surat_berharga	113673900	4246943	40	Karyawan_Swasta	3	4546912	93,4	Pria	Menikah	Tidak_ada_pinjaman	Bermasalah
Memiliki	60	Surat_berharga	50000000	1270834	32	PNS	2	4388072	29,64	Pria	Menikah	lunas	Lancar
Memiliki	60	Surat_berharga	128642160	3269655	32	PNS	2	4388072	76,25	Pria	Menikah	lunas	Bermasalah
Memiliki	36	Surat_berharga	38000000	1356389	39	PNS	2	4076200	33,28	Wanita	Menikah	lunas	Lancar
Memiliki	48	Surat_berharga	40000000	1183334	39	PNS	2	3690800	32,06	Wanita	Menikah	lunas	Lancar
Memiliki	36	Surat_berharga	27000000	963750	39	PNS	2	3993100	24,14	Wanita	Menikah	lunas	Lancar
Tidak_Memiliki	36	Surat_berharga	25000000	934028	40	Karyawan_Swasta	2	2603693	35,87	Wanita	Menikah	Tidak_ada_pinjaman	Lancar

Setelah dilakukan ujicoba terhadap 10 % data *testing* yang berjumlah 68 record, maka diperoleh hasil dari ujicoba sebagai berikut :

### GRAFIK HASIL PENGUJIAN *PROTOTYPE*



Gambar 4.25 Grafik hasil Uji Coba *Prototype*

Dari gambar 4.25 dapat disimpulkan bahwa *prototype* aplikasi penentuan kelayakan kredit sudah lebih dari cukup untuk membantu dalam memutuskan kelayakan dalam pemberian kredit, dimana 86 % hasil pemrosesan *prototype* aplikasi kelayakan pemberian kredit sesuai dengan data *testing* dan hanya 14 % hasil yang tidak sesuai.

#### 4.5 Implikasi Penelitian

Tidak semua kasus atau permasalahan harus diselesaikan dengan satu algoritma pada *data mining*. Karena belum tentu algoritma yang digunakan merupakan algoritma yang paling akurat. Oleh karena itu untuk menentukan algoritma yang paling akurat ini perlu dilakukan komparasi beberapa algoritma. Pada kasus penentuan kelayakan kredit ini menggunakan tiga algoritma yang dikomparasikan. Algoritma-algoritma itu adalah Algoritma C4.5, *Support Vector Machine*, dan *Neural Network*. Untuk dapat menentukan algoritma terbaik atau yang paling akurat maka akan dilakukan pengujian dengan *Cross Validation*, *Confusion Matrix*, *ROC Curve* dan *T-Test*. Dari pengujian ini akan didapatkan hasil algoritma yang paling akurat sehingga dapat disimpulkan bahwa algoritma tersebut layak untuk digunakan dalam membuat keputusan pada kelayakan penentuan kredit.

Hasil dari penelitian ini mempunyai implikasi terhadap penilaian kredit yaitu meliputi implikasi terhadap aspek sistem pendukung keputusan penilaian kredit, terhadap aspek manajerial dan terhadap aspek penelitian-penelitian selanjutnya yang akan diuraikan dibawah ini.

1. Implikasi terhadap aspek sistem pendukung keputusan penilaian kredit

Hasil menunjukkan bahwa metode *Neural Network* berbasis *Particle Swarm Optimization* merupakan metode yang cukup baik dalam pengklasifikasian data. Dengan demikian metode *Neural Network* berbasis *Particle Swarm Optimization* dapat memberikan pemecahan permasalahan analisa penilaian kredit dan dapat mendukung pengambilan keputusan dan pengembangan sistem informasi manajemen pada lembaga keuangan dan perbankan. Untuk menjalankan sistem dengan menggunakan algoritma *Neural Network* berbasis *Particle Swarm Optimization* dataset yang akan di input kedalam sistem perlu dilakukan tranformasi kebentuk yang lebih kecil untuk lebih mempercepat proses sistem dalam menentukan keputusan.

2. Implikasi terhadap aspek manajerial

Dari hasil penelitian ini diketahui bahwa metode *Neural Network* berbasis *Particle Swarm Optimization* dapat mendukung pengambilan keputusan dan pengembangan sistem informasi manajemen pada lembaga keuangan dan perbankan dengan menggunakan bantuan *software* RapidMiner, untuk itu diperlukan peningkatan kemampuan manajerial dari seorang manajer dan juga karyawan yang bersangkutan agar mampu membuat perencanaan secara formal, mengerjakan dan mengoperasikan sistem dengan baik dan benar. Hal tersebut dapat dilakukan dengan pelatihan atau training.

3. Implikasi terhadap aspek penelitian lanjutan

Dalam penelitian ini menggunakan metode *Neural Network* berbasis *Particle Swarm Optimization*, untuk mengetahui kehandalan metode, maka pada penelitian selanjutnya dapat dilakukan penggunaan data set lebih dari satu. Penelitian semacam ini dapat dikembangkan pada unit bisnis serupa atau yang lain. Pengembangan dapat dilakukan dengan pemilihan parameter menggunakan metode *Genetic Algorithm* untuk meningkatkan hasil optimasi. Penggunaan algoritma yang lain juga dapat digunakan misalkan dengan metode *Naive Bayes*, *K- Nearest Neighbor* dan sebagainya. Karena lembaga keuangan semakin berkembang maka kajian semacam ini dapat dilakukan secara periodik.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Dari penelitian yang telah dilakukan pembuatan menggunakan algoritma C4.5, *Support Vector Machine* dan *Neural Network* dengan berbasis *Particle Swarm Optimazation* dan menggunakan data perkreditan Bank DKI Jakarta Cabang Bendungan Hilir. Model yang dihasilkan dikomparasi untuk dapat diketahui metode terbaik dalam penentuan kelayakan pemberian kredit

Dari hasil pengujian dengan mengukur kinerja ketiga metode tersebut menggunakan *confusion matrix*, kurva *ROC* dan T-Test diketahui bahwa C4.5 menghasilkan nilai akurasi 93,81 % dan nilai AUC 0,941. Metode C4.5 berbasis PSO menghasilkan nilai akurasi 94,29 % dan nilai AUC 0,936. Metode *Support Vector Machine* menghasilkan nilai akurasi 89,85 % dan nilai AUC 0,932. Metode *Support Vector Machine* berbasis PSO menghasilkan nilai akurasi 96,20 % dan nilai AUC 0,989. Metode *Neural Network* menghasilkan nilai akurasi 93,83 % dan nilai AUC 0,944. Metode *Neural Network* berbasis PSO menghasilkan nilai akurasi 96,67 % dan nilai AUC 0,965. Nilai akurasi tertinggi yaitu pada algoritma *Neural Network* berbasis PSO sedangkan nilai AUC tertinggi yaitu pada algoritma *Support Vector Machine* berbasis PSO, oleh karena itu dilakukan proses pengujian T-Test dan menghasilkan algoritma *Neural Network* berbasis PSO lebih dominan dibandingkan algoritma *Support Vector Machine* berbasis PSO.

Dengan demikian metode *Neural Network* berbasis *Particle Swarm Optimazation* adalah metode yang terbaik untuk pemecahan masalah kelayakan penentuan kredit.

#### 5.2. Saran

Untuk keperluan penelitian lebih lanjut mengenai komparasi metode klasifikasi data mining dengan menggunakan data dibidang kredit perbankan dapat dilakukan dengan :

1. Menggunakan dataset lebih dari satu institusi untuk mengetahui keakuratan dari algoritma yang akan diuji.
2. Melakukan training kepada seluruh karyawan agar dapat dengan baik menjalankan sistem untuk mendapatkan keputusan dengan cepat dan tepat.
3. Melakukan pengujian dataset kredit dengan algoritma lain seperti metode *Naive Bayes*, *K- Nearest Neighbor* dan sebagainya, serta melakukan optimasi untuk peningkatan akurasi dengan algoritma *Genetic Algorithm*.

## Lampiran 1 Surat Keterangan Riset

**BANK DKI**  
PT BANK DKI JAKARTA Cab. Bendungan Hilir  
Jl. Bendungan Hilir Raya No. 1A Jakarta Pusat 10210, Jakarta Pusat

**SURAT PERNYATAAN**  
Nomor : 22/01/53/DKI/BH/I/2015

Yang bertanda tangan di bawah ini:

Nama : **Beni Samsumirat, SE**  
Alamat : Jl. Bendungan Hilir Raya No. 1A Jakarta Pusat 10210  
Jabatan : Bagian Resiko Kredit

Menerangkan dengan sebenarnya bahwa :

Nama : **Syaifur Rahmatullah A.R**  
NIM : 14000529  
Alamat : Jl. Jagakarsa 1 No. 7 Rt 002/02 Jakarta Selatan 12620

Adalah benar-benar telah melaksanakan penelitian di Bank DKI Cabang Bendungan Hilir Jakarta Pusat terhitung sejak 03 November s/d 24 November 2014 dalam rangka penyusunan Thesis dengan judul:

***“Komparasi Algoritma C4.5, Support Vector Machine, dan Neural Network berbasis Particle Swarm Optimazation Untuk Penentuan Kelayakan Kredit ”***

Demikian surat keterangan ini kami buat dengan sebenarnya, dan untuk dipergunakan sebagaimana mestinya.

Jakarta, 6 Januari 2015  
Bagian Resiko Kredit

  
**BANK DKI**  
BANK CABANG BENDUNGAN HILIR  
**Beni Samsumirat SE**

**Lampiran 2 Contoh Kuisisioner ISO 9126**

**Kuisisioner Sistem Informasi Penentuan  
Kelayakan Kredit**

Pekerjaan : *Risiko Kredit*.....

Umur : *28 tahun*.....

Pendidikan Terakhir : *Sistem Satu (SI)*.....

NO	PARAMETER	SETUJU	TIDAK SETUJU
1	Aplikasi yang ada apakah dapat dijalankan di lingkungan Komputer yang lain? (Portability)	✓	
2	Apakah Aplikasi yang ada dapat menjalankan Fungsinya sebagai aplikasi prediksi Penentuan kelayakan kredit pada nasabah? (Functionality)	✓	
3	Kehandalan aplikasi dalam menjalankan model uji coba penentuan kelayakan kredit telah terpenuhi? (reliability)	✓	
4	Apakah aplikasi yang ada sangat mudah digunakan? (usability)	✓	
5	Apakah penggunaan aplikasi mengefisienkan pekerjaan dalam penentuan pemberian kredit? (Efficiency)	✓	
6	Aplikasi yang ada sangat mudah untuk di modifikasi ? (Maintanability)		✓

**BANK ODKI**  
KANTOR CABANG BENDILANGAN HI

## Lampiran 3 Data Asli Nasabah Kredit

	akun_bank	lama_pinjam	jaminan	jumlah_kredit	jumlah_angsuran	umur	pekerjaan	masa_kerja	gaji	persentase_pemotongan_gaji	jenis_kelamin	status	pinjaman_sebelumnya	response
1	Memiliki	96	Surat_bertarga	108000000	2205000	45	PNS	4	5095000	43,28	Pria	Menikah	lunas	Lancar
2	Memiliki	77	Surat_bertarga	129000000	2911575	53	Guru	4	684750	42,72	Pria	Menikah	lunas	Lancar
3	Memiliki	36	Surat_bertarga	80000000	2888889	50	Guru	4	6924480	41,72	Wanita	Menikah	lunas	Lancar
4	Memiliki	48	Surat_bertarga	100000000	3000000	47	Guru	4	7289959	41,16	Wanita	Menikah	lunas	Lancar
5	Memiliki	84	Surat_bertarga	110000000	2409524	41	Guru	4	6713616	35,89	Wanita	Menikah	lunas	Lancar
6	Memiliki	87	Surat_bertarga	190000000	4044325	53	Guru	4	8271147	48,9	Wanita	Menikah	lunas	Lancar
7	Memiliki	84	Surat_bertarga	112000000	2453334	45	Guru	4	6385016	38,42	Wanita	Menikah	lunas	Lancar
8	Memiliki	96	Surat_bertarga	108000000	2205000	45	PNS	4	6615500	33,53	Pria	Menikah	lunas	Lancar
9	Memiliki	36	Surat_bertarga	100000000	3569445	50	Guru	4	9048300	39,45	Pria	Menikah	lunas	Lancar
10	Memiliki	79	Surat_bertarga	178000000	3996082	53	PNS	4	9080761	44,01	Pria	Menikah	lunas	Lancar
11	Memiliki	79	Surat_bertarga	272422830	6115864	50	PNS	4	9180761	67,35	Pria	Menikah	lunas	Bermasalah
12	Memiliki	60	Surat_bertarga	100000000	2541667	49	Guru	4	7939050	32,01	Pria	Menikah	lunas	Lancar
13	Memiliki	36	Surat_bertarga	60000000	2166667	53	Guru	4	8190668	26,56	Pria	Menikah	lunas	Lancar
14	Memiliki	36	Surat_bertarga	120000000	4333334	54	Guru	4	9757350	44,41	Pria	Menikah	lunas	Lancar
15	Memiliki	27	Surat_bertarga	89000000	4037963	58	Guru	4	8171350	49,42	Wanita	Menikah	lunas	Lancar
16	Memiliki	36	Surat_bertarga	70000000	2498612	48	Guru	4	6668984	37,81	Wanita	Menikah	lunas	Lancar
17	Memiliki	60	Surat_bertarga	80000000	2066667	48	PNS	4	5033236	41,06	Pria	Menikah	lunas	Lancar
18	Memiliki	60	Surat_bertarga	150000000	3900758	44	PNS	4	5133236	77,5	Pria	Lajang	Tidak_ada_pinjaman	Bermasalah
19	Memiliki	84	Surat_bertarga	100000000	2190477	47	Guru	4	6326240	34,63	Pria	Menikah	lunas	Lancar
20	Memiliki	84	Surat_bertarga	189000000	4157244	44	Guru	4	6226240	65,71	Pria	Menikah	belum_lunas	Bermasalah
21	Memiliki	60	Surat_bertarga	80000000	3066667	50	Guru	4	7088525	29,11	Wanita	Menikah	lunas	Lancar
22	Memiliki	60	Surat_bertarga	150000000	3487500	54	Guru	4	7393451	47,17	Wanita	Menikah	lunas	Lancar
23	Memiliki	60	Surat_bertarga	150000000	3875000	52	Guru	4	7393451	52,41	Wanita	Menikah	belum_lunas	Bermasalah
24	Memiliki	60	Surat_bertarga	50000000	1270834	44	Guru	3	6707296	18,95	Wanita	Menikah	lunas	Lancar
25	Memiliki	20	Surat_bertarga	48000000	2800000	54	Guru	4	6721934	41,65	Wanita	Menikah	lunas	Lancar
26	Memiliki	36	Surat_bertarga	55000000	1963195	47	Guru	4	7784986	25,22	Wanita	Menikah	lunas	Lancar
27	Memiliki	53	Surat_bertarga	99000000	2754175	51	Guru	4	5869300	46,58	Pria	Menikah	lunas	Lancar
28	Memiliki	48	Surat_bertarga	77000000	2261875	56	Guru	4	7510480	30,12	Pria	Menikah	lunas	Lancar

### Lampiran 4 Data Nasabah Kredit Transform

	akun_bank	lama_pinjam	jaminan	jumlah_kredit	jumlah_angsuran	umur	pekerjaan	masa_kerja	gaji	persentase_pemotongan_gaji	jenis_kelamin	status	pinjaman_sebelumnya	response
1	2	0,72	1	0,40	0,26	0,55	2	4	0,26	0,33	1	2	3	Lancar
2	2	0,58	1	0,46	0,33	0,70	5	4	0,32	0,32	1	2	3	Lancar
3	2	0,28	1	0,31	0,32	0,65	5	4	0,32	0,32	2	2	3	Lancar
4	2	0,37	1	0,37	0,33	0,59	5	4	0,34	0,31	2	2	3	Lancar
5	2	0,64	1	0,40	0,28	0,48	5	4	0,32	0,28	2	2	3	Lancar
6	2	0,66	1	0,65	0,43	0,70	5	4	0,37	0,36	2	2	3	Lancar
7	2	0,64	1	0,41	0,28	0,55	5	4	0,30	0,29	2	2	3	Lancar
8	2	0,72	1	0,40	0,26	0,55	2	4	0,31	0,26	1	2	3	Lancar
9	2	0,28	1	0,37	0,38	0,65	5	4	0,40	0,30	1	2	3	Lancar
10	2	0,60	1	0,61	0,42	0,70	2	4	0,40	0,33	1	2	3	Lancar
11	2	0,60	1	0,90	0,61	0,65	2	4	0,41	0,49	1	2	3	Bermasalah
12	2	0,46	1	0,37	0,29	0,63	5	4	0,36	0,25	1	2	3	Lancar
13	2	0,28	1	0,25	0,26	0,70	5	4	0,37	0,21	1	2	3	Lancar
14	2	0,28	1	0,44	0,45	0,72	5	4	0,43	0,33	1	2	3	Lancar
15	2	0,22	1	0,34	0,43	0,79	5	4	0,37	0,37	2	2	3	Lancar
16	2	0,28	1	0,28	0,29	0,61	5	4	0,31	0,29	2	2	3	Lancar
17	2	0,46	1	0,31	0,25	0,61	2	4	0,26	0,31	1	2	3	Lancar
18	2	0,46	1	0,53	0,41	0,54	2	4	0,26	0,55	1	1	1	Bermasalah
19	2	0,64	1	0,37	0,26	0,59	5	4	0,30	0,27	1	2	3	Lancar
20	2	0,64	1	0,65	0,44	0,54	5	4	0,30	0,48	1	2	2	Bermasalah
21	2	0,46	1	0,31	0,25	0,65	5	4	0,33	0,23	2	2	3	Lancar
22	2	0,46	1	0,48	0,38	0,72	5	4	0,34	0,35	2	2	3	Lancar

## DAFTAR RIWAYAT HIDUP

### A. Biodata Mahasiswa

NIM : 14000529  
Nama Lengkap : Syaifur Rahmatullah Abdul Rozak  
Tempat / Tanggal Lahir : Jakarta, 20 Oktober 1988  
Alamat Lengkap : Jl. Jagakarsa 1 No. 7 RT 02/002 Jak - Sel

### B. Pendidikan Formal dan Non-Formal

1. Tahun 2006 Lulus SMA, SMAN 38 Jakarta.
2. Tahun 2009 Lulus Diploma 3, Jurusan Manajemen Informatika, AMIK BSI Jakarta.
3. Tahun 2011 Lulus Sarjana, Jurusan Sistem Informasi, STMIK Nusa Mandiri

### C. Riwayat Pengalaman Berorganisasi / Pekerjaan

1. Anggota Pendiri Himpunan Mahasiswa Manajemen Informatika (HIMMI) AMIK BSI tahun 2008.
2. Staff Pengajar di AMIK BSI, Maret 2010 sampai dengan Sekarang

Jakarta, 1 Maret 2015

Hormat Saya,

**Syaifur Rahmatullah Abdul Rozak**