



TUGAS AKHIR - SM141501

**PERAMALAN KURS EUR/USD PADA PASAR *FOREX*
ONLINE MENGGUNAKAN *SUPPORT VECTOR*
REGRESSION YANG DIOPTIMASI DENGAN ALGORITMA
*PARTICLE SWARM OPTIMIZATION***

**ADE RHINALDY
NRP 1211 100 106**

**Dosen Pembimbing
PROF. DR. M. ISA IRAWAN, MT**

**JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - SM141501

**FORECASTING FOREIGN EXCHANGE RATE EUR/USD
BASED ON SUPPORT VECTOR REGRESSION
OPTIMIZED BY PARTICLE SWARM OPTIMIZATION**

**ADE RHINALDY
NRP 1211 100 106**

**Supervisor
PROF. DR. M. ISA IRAWAN, MT**

**Department Of Mathematics
Faculty Of Mathematics and Sciences
Sepuluh Nopember Institute Of technology
Surabaya 2015**

LEMBAR PENGESAHAN

PERAMALAN KURS EUR/USD PADA PASAR *FOREX* *ONLINE* MENGGUNAKAN *SUPPORT VECTOR* *REGRESSION* YANG DIOPTIMASI DENGAN *ALGORITMA PARTICLE SWARM OPTIMIZATION*

FORECASTING FOREIGN EXCHANGE RATE EUR/USD *BASED ON SUPPORT VECTOR REGRESSION* *OPTIMIZED BY PARTICLE SWARM OPTIMIZATION*

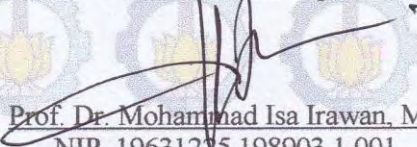
TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Sains
Pada Bidang Studi Matematika Terapan
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

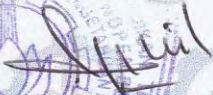
Oleh :

ADE RHINALDY
NRP. 1211 100 106

Menyetujui,
Dosen Pembimbing I,


Prof. Dr. Mohammad Isa Irawan, MT
NIP. 19631215 198903 1 001

Mengetahui,
Ketua Jurusan Matematika
FMIPA ITS


Prof. Dr. Erna Apriliani, M.Si
NIP. 19660414 199102 2 001

Surabaya, Juli 2015

**PERAMALAN KURS EUR/USD PADA PASAR FOREX
ONLINE MENGGUNAKAN *SUPPORT VECTOR
REGRESSION* YANG DIOPTIMASI DENGAN
ALGORITMA *PARTICLE SWARM OPTIMIZATION***

Nama : Ade Rhinaldy
NRP : 1211100106
Jurusan : Matematika FMIPA-ITS
Pembimbing : Prof. Dr. M. Isa Irawan, MT

Abstrak

Pasar *forex online* merupakan pasar uang yang memiliki tingkat likuiditas yang tinggi. Hal ini menyebabkan harga kurs EUR/USD pada pasar *forex online* bergerak fluktuatif. Para pelaku pasar *forex* memanfaatkan hal ini untuk mendapatkan keuntungan dengan cara melakukan posisi jual pada saat harga naik dan posisi beli pada saat harga turun. Untuk memudahkan trader dalam mengambil keputusan jual dan beli dibutuhkan peramalan data harga kurs EUR/USD. Penelitian ini bertujuan untuk memprediksi arah pergerakan harga harian perdagangan *forex* dengan menggunakan *support vector regression* (SVR). Parameter pada SVR di optimasi dengan menggunakan algoritma *particle swarm optimization* (PSO) sehingga didapatkan hasil peramalan yang baik. Hasil dari penelitian ini didapatkan hasil peramalan dengan nilai MAPE = 0.1895% yang dihasilkan dengan memasukkan nilai parameter optimal *penalty* dari SVR (C) = 102.2522 dan parameter kernel RBF (σ) = 1.9724.

Kata kunci: Peramalan, *forex*, *Support Vector Regression*, *Particle Swarm Optimization*.

“Halaman ini sengaja dikosongkan”

**FORECASTING FOREIGN EXCHANGE RATE EUR/USD
BASED ON SUPPORT VECTOR REGRESSION OPTIMIZED
BY PARTICLE SWARM OPTIMIZATION**

Name : Ade Rhinaldy
NRP : 1211100106
Department : Matematika FMIPA-ITS
Supervisor : Prof. Dr. M. Isa Irawan, MT

Abstract

The forex market is an online money market that has high liquidity. This led to the exchange rates EUR / USD on the online forex market moving to fluctuate. The forex market participants use this to get benefit by doing sell positions when price rises and buy positions when price fall. To facilitate traders take buy and sell decisions then forecasting of data exchange rates EUR / USD is required. Forecasting is one of mathematics branch that can be used as a tool. This study has purpose to predict the direction of daily forex price movement using support vector regression (SVR). Parameter on SVR is optimized by Particle Swarm Optimization (PSO) to obtain a good prediction. This study has results of forecasting with MAPE= 0.1895% which is produced by inserting the optimal parameter penalty from SVR (C) = 102.2522 and parameter kernel RBF (σ) = 1.9724.

Keywords: Forecasting, forex, Support Vector Regression, Particle Swarm Optimization.

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Alhamdulillahirabbil'aalamiin, Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat limpahan rahmat dan karunia-Nya penulis dapat menyusun Tugas Akhir dengan judul “Peramalan Kurs Eur/Usd Pada Pasar Forex Online Menggunakan *Support Vector Regression* Yang Dioptimasi Dengan Algoritma *Particle Swarm Optimization*” dengan baik dan tepat pada waktunya.

Dalam mengerjakan Tugas Akhir ini penulis menyampaikan ucapan terima kasih kepada seluruh pihak yang telah mendukung dan membantu sehingga Tugas Akhir ini bisa terselesaikan. Adapun pihak-pihak yang terlibat antara lain:

1. Ibu Prof. Dr. Erna Apriliani, M.Si sebagai Ketua Jurusan Matematika FMIPA ITS.
2. Bapak Prof. Dr. M.Isa Irawan, MT sebagai dosen pembimbing Tugas Akhir. Atas segala bimbingan, dukungan dan motivasi yang telah diberikan pada penulis.
3. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si, MT, Ibu Endah Rokhmah M.P., Ph.D, Bapak Dr. Chairul Imron, MI.Komp, Bapak Drs. Daryono Budi Utomo, M.Si, selaku dosen penguji.
4. Bapak Dr. Chairul Imron, MI.Komp sebagai Ketua Program Studi Sarjana Matematika Jurusan Matematika FMIPA ITS
5. Bapak dan Ibu dosen serta staff tata usaha dan laboratorium Jurusan Matematika FMIPA-ITS.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari semua pihak untuk perbaikan dimasa yang akan datang. Semoga laporan ini bermanfaat bagi semua pihak.

Surabaya, Agustus 2015

Penulis

Special Thanks to :

1. Ayah dan Alm. Ibu tercinta yang selalu ikhlas mendoakan anak-anaknya dalam menjalani kehidupan di dunia ini sehingga bisa menjadi pribadi yang beriman dan penuh manfaat.
2. Kedua saudaraku, mas Mawan dan Ridho yang selalu menyemangati penulis dalam menjalani hidup untuk menjadi lebih baik.
3. Seluruh keluarga besar penulis yang selalu memberi dukungan dan doanya.
4. Teman hidup Tyara Herdha, yang selalu ada untuk membantu dan menyemangati penulis disaat penulis mengalami kesulitan dalam menyelesaikan buku tugas akhir ini.
5. Teman-teman “Bodo Amat” Irvan, Yongky, Dedek, Raffy, Syukron, Fiyan, Ibing, Putri, Kusuma dan Intan yang selalu mendukung dan membantu untuk menyelesaikan tugas akhir ini.
6. Teman-teman kosan, Biofanda dan Adit yang selalu bersedia untuk berbagi canda dan tawa ketika lelah dalam mengerjakan tugas akhir.
7. Teman-teman begadang di lab. komputasi, Lusi, Dina koboy, Veda, Liyana, Riyani, Hakam, Habib, Yahya dll.
8. Teman-teman HIMATIKA ITS kepengurusan periode 2013/2014, yang telah memberikan pengalaman yang luar biasa.
9. OMITS *Crew* 2014, yang telah memberikan kenangan dn pengalaman yang tidak akan terlupa.
10. Teman-teman seperjuangan dalam pengkaderan HIMATIKA ITS kelompok Euclid, Yongky, Handy, Henny, Nuril, Ainur, Hurry, Christin, Ifa, dan Eni.
11. Orang-orang yang membantu memecahkan masalah dalam tugas akhir ini, Mas Elsen, Mas Ipin dan Mas Satria

DAFTAR ISI

	Hal
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR SIMBOL	xix
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Laporan	4
BAB II TINJAUAN PUSTAKA	
2.1 Pasar Forex Online	7
2.2 Peramalan.....	9
4.2.1 Peramalan Kualitatif	9
4.2.2 Peramalan Kuantitatif	10
2.3 <i>Support Vector Regression</i> (SVR)	10
2.4 <i>Particle Swarm Optimization</i> (PSO).....	15
2.5 Normalisasi Data	18
2.6 Nilai Keakuratan Peramalan.....	19
BAB III METODOLOGI PENELITIAN	
3.1 Studi Literatur	21
3.2 Pengumpulan Data Peramalan.....	21
3.3 Variabel Penelitian	21

3.4	Peramalan dengan Metode SVR	22
3.5	Peramalan dengan Metode PSO-SVR	23
3.6	Analisis Pengolahan Data dan Penarikan Kesimpulan ..	24
BAB IV ANALISIS DAN PEMBAHASAN		
4.1	Data Observasi	29
4.2	Prapemrosesan	29
4.4.1	Pembagian Data	29
4.4.2	Penyusunan Data	30
4.3	Peramalan Data Menggunakan Metode SVR.....	31
4.4	Peramalan Data Menggunakan Metode PSO-SVR	36
4.4.1	<i>Training</i> Data	37
4.4.2	<i>Testing</i> Data	45
BAB V PENUTUP		
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA		51
LAMPIRAN		
1.	Data <i>Time Series Closing Price</i> kurs EUR/USD	53
2.	Data Ternormalisasi	54
3.	Bobot dan Bias Metode SVR	59
4.	Bobot dan bias Metode PSO-SVR	68
5.	<i>Listing</i> Program PSO-SVR	69
6.	<i>Listing</i> Program <i>Training</i> SVR	71
7.	<i>Listing</i> Program <i>Testing</i> SVR	73
8.	<i>Listing</i> Program Normalisasi	73
9.	<i>Listing</i> Program Evaluasi <i>Fitness</i>	73
10.	<i>Listing</i> Program Update Kecepatan Partikel	74
BIODATA PENULIS		75

DAFTAR TABEL

Tabel 2.1 Skala Akurasi Peramalan	21
Tabel 3.1 Penyusunan Data	22
Tabel 4.1 Keakuratan Model SVR	35
Tabel 4.2 Nilai Awal Parameter	38
Tabel 4.3 Nilai Kelayakan (<i>current</i> fitness) Iterasi 0	39
Tabel 4.4 Nilai Px_best C , Px_best σ dan Pf_best	40
Tabel 4.5 Kecepatan Partikel pada Iterasi 1	41
Tabel 4.6 Posisi Partikel Yang Dipebarui	41
Tabel 4.7 Partikel dan MAPE Konvergen Pada Suatu Nilai ...	33
Tabel 4.8 Perbandingan Data Aktual Dan Data Peramalan ...	45

“Halaman ini sengaja dikosongkan”

DAFTAR SIMBOL

Support Vector Regression

x_i	: Data input
y_i	: <i>Output</i> atau target
w	: Bobot
b	: Bias
ξ_i	: Variabel <i>slack</i>
α_i	: Pengali lagrange
K	: Kernel
σ	: parameter kernel RBF

Particle Swarm Optimization

N	: Jumlah Partikel
X	: Partikel
$X^{(B)}$: Batas bawah
$X^{(A)}$: Batas atas
$X(i)j$: Posisi partikel j pada iterasi i
$V(i)j$: Kecepatan partikel j pada iterasi i
c_1	: <i>Learning rates</i>
c_2	: <i>Learning rates</i>

Nilai Keakuratan Peramalan

e_i	: Nilai eror
X_i	: Data aktual
F_i	: Peramalan
n	: Banyak data

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Valuta asing (valas) atau *foreign exchange (forex)* diartikan sebagai mata uang asing dan alat pembayar lainnya yang digunakan untuk melakukan atau membiayai transaksi ekonomi keuangan internasional dan mempunyai catatan kurs resmi pada bank sentral [1]. Perdagangan valas atau *forex* merupakan kegiatan perdagangan mata uang dari negara yang berbeda satu sama lain. Pada pasar *forex* yang diperdagangkan adalah mata uang dengan pelakunya yang sangat luas meliputi bank, perusahaan besar, negara, institusi, spekulasi bahkan perorangan. Besarnya pelaku perdagangan pasar *forex* yang mencakup seluruh negara yang ada di dunia menjadikan pasar *forex* sebagai pasar uang yang paling besar dan sangat likuid. Pelaku pasar *forex* yang ada di seluruh dunia ini juga menyebabkan pasar *forex* buka 24 jam tanpa berhenti selama lima hari dalam seminggu. Sejak pertengahan tahun 1980, perdagangan *forex* atau trading *forex* mulai dikerjakan melalui jaringan perdagangan elektronik. Media perdagangan *forex* terus berkembang hingga saat ini transaksi perdagangan di pasar *forex* telah dapat dilakukan secara *online* yang biasa disebut dengan era *forex online*. Kemajuan ini menyebabkan proses transaksi pada pasar *forex* terjadi lebih cepat, dimanapun, kapanpun, dan dengan jumlah yang sangat besar.

Pasar *forex* merupakan pasar yang memiliki kelebihan di tingkat likuiditas yang tinggi pada waktu ke waktu. Tingginya likuiditas menyebabkan harga di pasar mata uang bergerak sangat fluktuatif. Hal ini menyebabkan *forex trading* menjadi investasi pada sektor finansial yang tergolong paling *high risk-high return investment*. Artinya, peluang untuk memperoleh keuntungan sangat besar bahkan dapat mencapai ratusan persen perbulan namun diimbangi dengan kemungkinan kerugian yang besar

apabila modal yang diinvestasikan oleh para trader tidak dikelola dengan baik.[2].

Tingginya fluktuasi harga *forex* menyebabkan para pedagang *forex* (*trader*) harus mengetahui waktu yang tepat untuk mengambil keputusan kapan melakukan penjualan dan pembelian. Untuk membantu para *trader* dalam menentukan perencanaan kapan membeli dan kapan menjual maka digunakan perhitungan peramalan.

Terdapat berbagai macam metode untuk mendapatkan hasil ramalan. Salah satu metode yang dapat digunakan adalah metode *support vector regression* (SVR) yang merupakan cabang dari *support vector machine* (SVM).

SVM muncul sebagai suatu teknik baru dan kuat dalam proses pembelajaran mesin (*machine learning*) dari data dan secara khusus dalam penyelesaian permasalahan klasifikasi dan SVR sebagai penyelesaian permasalahan regresi [3]. SVR merupakan metode pembelajaran mesin yang sangat cocok untuk penyelesaian permasalahan dengan data yang memiliki tingkat fluktuasi data tinggi [3]. Tingginya fluktuasi data yang dimunculkan pada pasar *forex* mendorong penulis memilih metode pembelajaran mesin SVR sebagai metode dalam penelitian ini.

Dewasa ini SVM dan SVR banyak digunakan dalam berbagai macam penelitian. Berny Pebo pada penelitian[4] menggunakan *multiclass twin bounded support vector machine* untuk pengenalan ucapan. Pada penelitian [5] yang dilakukan Heru Supriyanto, di dapatkan hasil peramalan harga harian valuta asing (EUR/USD, GBP/USD, dan USD/JPY) dengan menggunakan SVM dimana parameter di optimasi dengan algoritma *gridsearch*. Sementara, Xibin Wang dkk. pada tulisan ilmiahnya [6] menyelesaikan permasalahan peramalan harga *real estate* dengan menggunakan SVM dimana parameter dioptimasi dengan algoritma *particle swarm optimization* (PSO). Xibin Wang dkk. dalam tulisan ilmiahnya [6] mengatakan teknik optimasi yang paling umum adalah algoritma koloni semut,

algoritma *gridsearch* dan algoritma genetika. Algoritma koloni semut memiliki kelemahan diantaranya langkanya feromon awal, waktu pencarian yang lama dan memiliki solusi terbaiknya bersifat lokal; algoritma *gridsearch* adalah algoritma yang bersifat komputasi intensif, memakan waktu yang lama dan akurasi pembelajar (*training*) rendah; algoritma genetika adalah algoritma yang operasinya kompleks dan dibutuhkan isu-isu yang berbeda untuk merancang *crossover* yang berbeda atau mutasi. PSO merupakan sebuah algoritma dengan konsep sederhana tetapi sangat efektif dalam mengoptimasi permasalahan berbagai fungsi, selain itu algoritma PSO mudah dalam implementasi dan memiliki konvergensi yang cepat [7].

Berdasarkan latar belakang diatas, maka penulis mengangkat judul “Peramalan Kurs EUR/USD Pada Pasar *Forex Online* Menggunakan *Support Vector Regression* yang Dioptimasi Dengan Algoritma *Particle Swarm Optimization*”.

1.2 Rumusan Masalah

Rumusan masalah dalam penulisan tugas akhir ini adalah:

1. Bagaimana mendapatkan parameter optimal dari metode SVR dengan menggunakan algoritma PSO.
2. Bagaimana peramalan kurs EUR/USD pada pasar *forex online* menggunakan SVR dengan parameter yang telah optimal dan analisis tingkat *error* dari hasil peramalan tersebut.

1.3 Pembatasan Masalah

Untuk memperjelas rumusan masalah pada tugas akhir ini, maka batasan masalahnya adalah :

1. Data yang digunakan adalah data *closing* kurs EUR/USD yang bergerak setiap enam puluh menit yang diambil selama 5×24 jam.
2. Kernel yang digunakan adalah kernel *Radial Basis Function* (RBF)

3. Parameter yang dioptimasi adalah parameter C (pinalti) dan σ (kernel RBF).
4. Perhitungan peramalan menggunakan Matlab.

1.4 Tujuan

Tujuan penelitian ini yaitu :

1. Mendapatkan parameter yang optimal dari metode SVR dengan menggunakan algoritma PSO.
2. Meramalkan kurs EUR/USD pada pasar *forex online* menggunakan SVR dengan parameter yang telah optimal dan menganalisis tingkat *error* dari hasil peramalan tersebut.

1.5 Manfaat

Manfaat yang diharapkan oleh penulis dari penulisan tugas akhir ini adalah

1. Sebagai alat bantu analisa bagi para pelaku pasar *forex* (*trader forex*) dalam mengambil keputusan beli dan jual.
2. Sebagai referensi dan informasi tentang penggunaan metode PSO - SVR dalam peramalan.

1.6 Sistematika Penulisan Laporan

Penulisan laporan tugas akhir ini disusun dalam lima bab sebagai berikut:

1. BAB I : Pendahuluan

Bab ini berisikan latar belakang dari penelitian penulis, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi penelitian, pelaksana kegiatan, tempat dan waktu pelaksanaan, serta sistematika penulisan.

2. BAB II : Tinjauan Pustaka

Bab ini menjabarkan tentang materi-materi yang berkaitan dengan pasar *forex online*, peramalan, metode SVR, dan optimasi parameter menggunakan PSO. Materi-materi tersebut digunakan sebagai acuan dalam mengerjakan tugas akhir.

3. BAB III : Metode Penelitian

Bab ini menjelaskan langkah – langkah pengerjaan perhitungan peramalan dengan menggunakan metode SVR yang parameternya dioptimasi dengan menggunakan algoritma PSO.

4. BAB IV : Analisis dan Pembahasan

Bab ini membahas hasil pengolahan data yang dihasilkan dari metode SVR dan algoritma PSO.

5. BAB V : Kesimpulan dan Saran

Bab ini menjelaskan tentang penarikan kesimpulan dan saran dari penelitian yang telah dilakukan.

“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan membahas landasan atas teori-teori yang bersifat ilmiah untuk mendukung penelitian ini. Ada dua tinjauan pustaka yang akan dibahas, yaitu pustaka terdahulu dan kajian pustaka. Keduanya akan membahas teori-teori ilmiah yaitu antara lain, pasar *forex online*, peramalan, metode SVR.

2.1 Pasar *Forex Online*

Pasar valuta asing (valas) atau dalam bahasa Inggris disebut *foreign exchange market (forex)* merupakan suatu jenis perdagangan atau transaksi yang memperdagangkan mata uang suatu negara terhadap mata uang negara lainnya (pasangan mata uang/pair) yang melibatkan pasar-pasar uang utama di dunia selama 24 jam dan lima hari dalam seminggu secara berkesinambungan. Sejarah *forex* telah dimulai pada tahun 1973 setelah terjadinya pertemuan Bretton Woods. Uang dikenal sebagai alat pertukaran dan pembayaran transaksi yang sah, menggantikan sistem emas yang telah ditinggalkan pada tahun 1971. Perdagangan yang meluas pada saat itu, yang tidak hanya terjadi didalam satu negara (lokal) namun juga secara global, menimbulkan masalah karena masing-masing mata uang tidak mempunyai nilai yang sama antara satu dengan lainnya [8]. Selepas tahun 1971 sistem Bretton Woods dihapus, hal ini dikarenakan terjadinya krisis Eropa yang berkepanjangan yang mengakibatkan terbelahnya hubungan perdagangan eropa dengan US. Setelah tidak dipakainya lagi sistem Bretton Woods dan melalui *Jamaica Agreement* tahun 1976, Transaksi *forex* mengalami peningkatan signifikan. Yaitu ketika nilai tukar mata uang antar negara sepenuhnya berada dibawah kendali mekanisme pasar. Dengan kata lain pasarlah yang akan menentukan apakah nilai tukar uang tersebut akan naik ataupun turun hingga saat ini [9].

Seperti halnya dalam sebuah perdagangan jual-beli biasanya, dalam perdagangan *forex* pun pada dasarnya membutuhkan kedua sisi yaitu pembeli dan penjual. Namun disini bedanya, pembeli dan penjual tidak pernah melakukan pertemuan fisik secara langsung dan tidak pernah terjadi serah terima secara fisik juga. Semuanya dilakukan dalam bentuk perjanjian dan diperantarai oleh pihak ketiga yang biasa disebut sebagai pialang atau broker. Tugas pialang adalah menjadi semacam pengumpul transaksi yang dilakukan secara *retail* oleh *investor-investor* di bawahnya untuk kemudian diteruskan kepada pasar atau bursa. Bursa disini adalah tempat pertemuan antara penjual dan pembeli dalam bertransaksi. Perbedaan bursa dengan pasar konvensional adalah pada bursa biasanya tidak terjadi transaksi jual beli secara retail perorangan tetapi biasanya di akumulasikan dan baru kemudian di eksekusi. Demikian mekanisme perdagangan *forex* dalam dunia sehari-hari.

Seiring perkembangan teknologi yang sangat pesat membuat transaksi di pasar *forex* sekarang bisa dilakukan dengan bantuan internet. Sehingga transaksi di pasar *forex* dapat dilakukan kapanpun dan dimanapun. *Platform trading forex online* berbasis internet pertama kali muncul pada tahun 1996. *Platform* ini memungkinkan perdagangan *forex* dengan mudah dan cepat. Para *trader* dapat mengakses pasar dan bertransaksi lewat komputer di rumah mereka sendiri. *Platform* yang banyak dipakai oleh para *trader* saat ini adalah metatrader yang mulai dibuat oleh perusahaan MetaQuotes Software pada tahun 2005. *Software* metatrader merupakan platform yang dilengkapi dengan berbagai macam fasilitas variasi grafik dan ratusan indikator analisis grafik. Hingga tulisan ini dibuat metatrader telah berkembang hingga dapat diakses dengan menggunakan *android*. Dalam penulisan ilmiah kali ini, data kurs EUR/USD pada pasar *forex* digunakan sebagai data yang akan diramal.

2.2 Peramalan

Peramalan adalah proses untuk memperkirakan berapa kebutuhan dimasa mendatang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan untuk memenuhi permintaan barang atau jasa [10]. Peramalan merupakan salah satu kajian matematika yang sangat sering kita jumpai kegunaannya. Banyak kegiatan manusia sehari-hari yang terbantu dengan adanya peramalan sebagai dasar dari perencanaan. Berdasarkan sifatnya peramalan terbagi menjadi dua yaitu [11]

2.2.1 Peramalan Kualitatif

Peramalan kualitatif merupakan peramalan yang bersifat subjektif yang ramalannya berdasarkan pengalaman, keahlian dan pendapat seseorang yang merupakan pakar di bidangnya. Ramalan kualitatif tidak berbentuk angka misalnya, besok akan turun hujan, bulan depan rupiah akan menguat terhadap dolar. Bebearapa model peramalan yang digolongkan sebagai model kualitatif adalah :

1. Dugaan Manajemen

Merupakan metode peramalan dimana peramalan semata-mata berdasarkan pertimbangan manajemen.

2. Riset Pasar

Merupakan metode peramalan berdasarkan hasil survey pasar yang dilakukan oleh tenaga pemasar produk atau yang mewakilinya.

3. Metode Kelompok Terstruktur

Merupakan metode peramalan berdasarkan proses konvergensi dari opini beberapa orang ahli secara interaktif dan membutuhkan fasilitator untuk menyimpulkan hasil dari peramalan.

4. Analogi Historis

Merupakan teknik *forecasting* berdasarkan pola data masa lalu dari produkyang disamakan secara analogi

2.2.2 Peramalan Kuantitatif

Peramalan Kuantitatif merupakan peramalan yang didasarkan atas data kuantitatif pada masa lalu. Metode peramalan kuantitatif dibagi menjadi dua yaitu :

1. Metode deret berkala (*Time Series*)

Merupakan metode kuantitatif yang didasarkan atas penggunaan analisa pola hubungan antar variabel yang akan diperkirakan dengan variabel waktu. Permintaan dimasa lalu pada analisa deret waktu akan dipengaruhi keempat komponen utama yaitu, *trend* (tren), *cycle* (siklus), *season* (musiman) dan *random* (acak).

2. Metode Kausal

Merupakan metode kuantitatif yang mengembangkan suatu model sebab-akibat antara permintaan yang diramalkan dengan variabel-variabel lain yang dianggap berpengaruh. Metode Kausal dibagi lagi menjadi tiga bagian yaitu, Metode korelasi regresi, metode *ekinimetrik* dan metode *input output*.

2.3 Support Vector Regression (SVR)

Cabang SVM dalam permasalahan regresi dan peramalan time series dikenal sebagai SVR. SVM merupakan salah satu model mesin pembelajaran yang relatif baru. Pada tahun 1992 SVM pertama kali dipresentasikan di *Annual Workshop on Computational Learning Theory* oleh Boser, Guyon dan Vapnik. SVM banyak menarik para peneliti untuk menggunakannya, karena tergolong metode yang masih baru dan memiliki kemampuan yang baik dalam pembelajaran mesin untuk permasalahan regresi dan klasifikasi. Dengan menggunakan konsep ϵ *insensitive loss function* yang diperkenalkan oleh Vapnik, SVM bisa digeneralisasi untuk melakukan pendekatan fungsi atau regresi [12]

Misalkan diberikan set data pembelajaran, (x_i, y_i) dimana $i = 1, \dots, n$ dengan data *input* $x = \{x_1, \dots, x_n\} \in \mathcal{R}^N$ dan data *output* $y = \{y_1, \dots, y_n\} \in \mathcal{R}$. Dengan SVR, kita hendak

menemukan suatu fungsi $f(x)$ yang mempunyai deviasi (ε) paling besar dari target aktual y_i untuk semua data *training*.

Model regresi mendefinisikan hubungan antara x_i dan $f(x_i)$ sebagai berikut [12] :

$$y = f(x_i) = w\varphi(x_i) + b \quad (2.1)$$

dimana $\varphi(x_i)$ menunjukkan suatu titik didalam *feature space* F hasil pemetaan x di dalam *input space*. Parameter w dan b merupakan parameter bobot dan bias dari persamaan regresi Persamaan (2.1). Nilai dari w dan b di estimasi dengan cara meminimalkan fungsi resiko (*risk function*) yang didefinisikan dalam permasalahan optimasi berikut [12]

$$\min\left(\frac{1}{2} \|w\|^2 + C \frac{1}{n} \sum_{i=1}^n L_{\varepsilon}(y_i, f(x_i))\right) \quad (2.2)$$

dengan kendala:

$$y_i - w\varphi(x_i) - b \leq \varepsilon$$

$$w\varphi(x_i) - y_i + b \leq \varepsilon, i = 1, \dots, n$$

dengan

$$\sum_{i=1}^n L_{\varepsilon}(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon & |y_i - f(x_i)| \geq \varepsilon \\ 0, & \text{untuk yang lain} \end{cases} \quad (2.3)$$

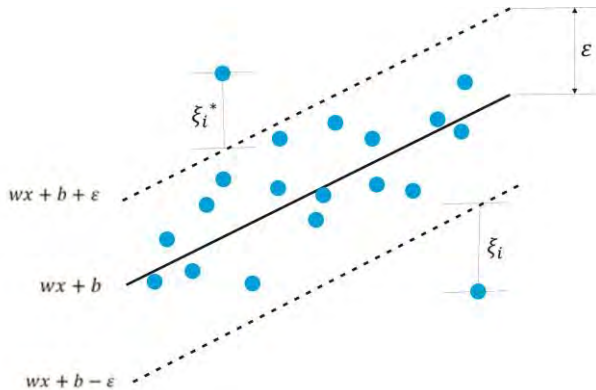
Faktor $\|w\|^2$ dinamakan *regularisasi*. Meminimumkan $\|w\|^2$ akan membuat suatu fungsi memiliki jarak yang saling dekat, sehingga bisa mengontrol kapasitas fungsi. Faktor kedua dalam fungsi

tujuan adalah kesalahan *empiric* yang diukur dengan ε -insensitive loss function. Agar mendapatkan generalisasi yang baik untuk fungsi regresi f maka $\|w\|$ harus diminimalkan dengan cara menyelesaikan permasalahan optimasi berikut :

$$\min \frac{1}{2} \|w\|^2 \quad (2.4)$$

kendala :

$$\begin{aligned} y_i - w\varphi(x_i) - b &\leq \varepsilon \\ w\varphi(x_i) + b - y_i &\leq \varepsilon \end{aligned}$$



Gambar 2.1 Bentuk model SVR

Diasumsikan ada suatu fungsi f yang dapat mengaproksimasi semua titik (x_i, y_i) dengan presisi ε . Dalam hal mengatasi permasalahan adanya beberapa titik data yang mungkin keluar dari rentang $f \pm \varepsilon$ digunakan variabel *slack* ξ_i dan ξ_i^* untuk mengatasi masalah pembatas yang tidak layak dalam problem optimasi. Adanya variabel *slack* menyebabkan penyelesaian pada permasalahan optimasi menjadi sebagai berikut :

$$\min_{w,b,\xi^*} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \right) \quad (2.5)$$

dengan kendala

$$y_i \leq ((w \cdot x_i) + b) + \varepsilon + \xi_i, \quad i = 1, \dots, n$$

$$y_i \geq ((w \cdot x_i) + b) - \varepsilon - \xi_i^*, \quad i = 1, \dots, n$$

$$\xi_i \geq 0, \quad \xi_i^* \geq 0$$

Dengan kata lain, penentuan nilai parameter w dan b menjadi masalah pemrograman kuadrat (*quadratic programming*), yaitu meminimumkan suatu fungsi kuadrat dengan syarat suatu pertidaksamaan linier.

Untuk menyelesaikan pemrograman kuadrat tersebut, cara yang umum digunakan adalah mencari bentuk dual dengan menggunakan perkalian lagrange (*lagrange multipliers*). Dengan menggunakan pengali lagrange α_i dan α_i' , dimana α_i dan α_i' merupakan variabel non-negatif, maka bentuk primal pemrograman kuadrat (2.5) dapat diubah menjadi bentuk dual sebagai berikut :

$$\max \left(\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i') (\alpha_i - \alpha_i') K(x_i x_j) \right) \quad (2.6)$$

$$+ \sum_{i=1}^l y_i ((\alpha_i - \alpha_i') - \varepsilon) - \sum_{i=1}^l (\alpha_i + \alpha_i') \xi_i^* \geq 0$$

dengan kendala

$$\sum_{i=1}^l (\alpha_i - \alpha_i') = 0$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, l$$

Dengan menggunakan lagrange *multiplier* dan kondisi optimalitas, fungsi regresi dirumuskan sebagai :

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i') K(x_i, x) + b \quad (2.7)$$

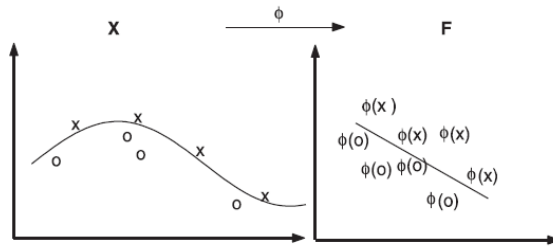
dimana $K(x_i, x)$ didefinisikan melalui fungsi kernel dan $w = (\alpha_i - \alpha_i')$.

Metode kernel [13] memberikan pendekatan alternatif dengan cara melakukan pemetaan data x dari *input space* ke *feature space* F melalui suatu fungsi ϕ sehingga $\phi: x \rightarrow \phi(x)$. Karena itu suatu titik x pada *input space* menjadi $\phi(x)$ di *feature space*. Pemetaan *input space* ke *feature space* dapat dilihat pada Gambar 2.2. Hal ini menyebabkan $K(x, x') = \phi(x) \cdot \phi(x')$ dimana (\cdot) merupakan operasi perkalian dalam. Kemudian kita bisa membuat suatu fungsi pemisah yang linier yang mewakili fungsi nonlinier di *input space*.

Penggunaan fungsi kernel yang berbeda akan menghasilkan model regresi yang berbeda pula. Fungsi kernel yang biasanya dipakai dalam literatur SVM [6]:

Kernel linier	: $K(x, x') = x^T x$
Polinomial	: $K(x, x') = (x^T x_i + 1)^p$
Kernel RBF	: $K(x, x') = \exp\left(\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right)$

Pada permasalahan peramalan kali ini, penulis menggunakan fungsi kernel RBF dengan σ sebagai parameter. Parameter σ mendefinisikan struktur tingkat dimensi *feature space*. Pada penjabaran di atas kita ketahui bahwa pemilihan dari parameter C dan σ memiliki pengaruh yang besar terhadap keakuratan peramalan. Oleh karena itu algoritma PSO digunakan untuk mendapatkan parameter dari SVR yang optimal.



Gambar 2.2 Pemetaan kernel mengubah permasalahan nonlinier menjadi linier

2.4 Particle Swarm Optimization (PSO)

PSO merupakan metode optimasi yang pertama kali dikembangkan oleh James Kennedy dan Russell Eberhart [7]. PSO merupakan metode optimasi dimana algoritmanya mengadopsi pada perilaku dan psikologi sosial sebuah organisme seperti semut, rayap, lebah atau burung [14]. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya.

PSO dikembangkan dengan berdasarkan pada model berikut, dalam hal ini partikelnya adalah burung [14]:

1. Ketika seekor burung mendekati target atau makanan (atau bisa meminimumkan atau memaksimumkan suatu fungsi tujuan) secara cepat mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu.
2. Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung
3. Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

Model ini akan terus melakukan iterasi dan di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi). Model ini disimulasikan dalam ruang dengan dimensi tertentu.

Diberikan permasalahan untuk meminimumkan fungsi $f(x)$ sebagai berikut:

$$\min f(x)$$

$$\text{dengan } X^{(B)} \leq X \leq X^{(A)}$$

Pada algoritma PSO, X merupakan populasi atau kelompok awal dengan rentang antara $X^{(B)}$ yang merupakan batas bawah dari populasi dan $X^{(A)}$ yang merupakan batas atas populasi. Dengan menggunakan algoritma PSO maka langkah-langkah pengerjaannya sebagai berikut [14] :

1. Diasumsikan bahwa ukuran populasi atau kelompok adalah sebanyak N partikel. Untuk mengurangi jumlah evaluasi fungsi yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar dan juga tidak terlalu kecil. Biasanya digunakan ukuran kawanan adalah 20 sampai 30 partikel.
2. Bangkitkan populasi awal X dengan rentang $X^{(B)}$ dan $X^{(A)}$ secara *random*, sehingga didapat X_1, X_2, \dots, X_N . Posisi partikel j dan kecepatannya pada iterasi i dinotasikan sebagai $X(i)j$ dan $V(i)j$. Sehingga partikel-partikel awal ini akan menjadi $X_1(0), X_2(0), \dots, X_N(0)$. Vektor $X_j(0)$, ($j = 1, 2, \dots, N$) disebut partikel atau vektor koordinat dari partikel. Evaluasi nilai fungsi tujuan untuk setiap partikel dan dinyatakan dengan

$$f|X_1(0)|, f|X_2(0)|, \dots, f|X_N(0)|$$

3. Hitung kecepatan dari semua partikel. Awalnya semua kecepatan dari partikel diasumsikan sama dengan nol. Set iterasi $i = 1$.
4. Pada iterasi ke- i , temukan 2 parameter penting untuk setiap partikel j yaitu :
 - a. Nilai terbaik sejauh ini dari $X_j(i)$ dan dinyatakan sebagai $P_{best,j}$ dengan nilai fungsi obyektif paling rendah (kasus minimasi) dan fungsi obyektif paling tinggi (kasus maksimum), $f|X_j(i)|$, yang ditemui sebuah partikel j pada semua iterasi sebelumnya. Nilai terbaik untuk semua partikel $X_j(i)$ yang ditemukan sampai iterasi ke- i , G_{best} , dengan nilai fungsi tujuan paling kecil/minimum diantara semua partikel untuk semua iterasi sebelumnya, $f|X_j(i)|$.
 - b. Hitung kecepatan partikel j pada iterasi ke i dengan persamaan sebagai berikut:

$$V_j(i) = V_j(i - 1) + c_1 r_1 [P_{best,j} - x_j(i - 1)] + c_2 r_2 [G_{best} - x_j(i - 1)] \quad (2.8)$$

$$j = 1, 2, \dots, N$$

dimana c_1 dan c_2 masing-masing adalah *learning rates* untuk kemampuan individu (*cognitive*) dan pengaruh sosial atau kelompok, r_1 dan r_2 bilangan *random* yang berdistribusi *uniform* dalam interval 0 dan 1. Jadi parameters c_1 dan c_2 menunjukkan bobot dari *memory* posisi sebuah partikel terhadap *memory* posisi dari kelompok (*swarm*). Nilai dari c_1 dan c_2 biasanya adalah 2 sehingga perkalian $c_1 r_1$ dan $c_2 r_2$ memastikan bahwa

partikel-partikel akan mendekati target sekitar setengah selisihnya.

- c. Hitung posisi atau koordinat partikel j pada iterasi ke- i dengan cara :

$$X_j(i) = X_j(i - 1) + V_j(i) \quad (2.9)$$

$$j = 1, 2, \dots, N$$

Evaluasi nilai fungsi tujuan untuk setiap partikel dan nyatakan sebagai :

$$f|X_1(i)|, f|X_2(2)|, \dots, f|X_N(i)|$$

5. Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah 4 diulang dengan memperbarui iterasi $i = i + 1$, dengan cara menghitung nilai baru dari $P_{best,j}$ dan G_{best} . Proses iterasi ini dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (*stopping criteria*), misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

2.5 Normalisasi Data

SVR bekerja lebih efektif dengan data yang mempunyai rentang skala kecil. Selain itu, normalisasi dapat mengurangi . Metode yang dipilih untuk menormalisasi data adalah metode *min-max*. Metode *min-max* ini dapat dilihat pada persamaan :

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \quad (2.10)$$

dengan :

v'_i	: data baru hasil normalisasi
v_i	: data input yang akan di normalisasi
\min_A	: nilai minimum dari data input
\max_A	: nilai maksimum dari data input
new_min_A	: nilai minimum dari data hasil normalisasi yang diinginkan
new_max_A	: nilai maksimum dari data hasil normalisasi yang diinginkan

2.6 Nilai Keakuratan Peramalan

Untuk mengetahui sebaik apa hasil ramalan yang dilakukan, maka dibutuhkan analisis *error* antara data ramalan dan data aktual. Semakin kecil *error* semakin baik hasil ramalan. Jika diberikan X_i data aktual pada periode i dan F_i data ramalan untuk periode yang sama, maka *error* didefinisikan sebagai berikut.

$$e_i = X_i - F_i \quad (2.11)$$

Tugas akhir ini menggunakan nilai tengah galat persentase absolut (*Mean Absolute Percentage Error*) sebagai alat untuk menganalisis tingkat *error* [11]. Semakin kecil nilai MAPE yang dihasilkan maka semakin baik dan akurat pula model tersebut untuk digunakan. Persamaan MAPE dapat dilihat pada Persamaan 2.12.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - F_i}{X_i} \right| \times 100\% \quad (2.12)$$

Berikut ini skala akurasi peramalan menurut Lewis[15]:

Tabel 2.1. Skala Akurasi Peramalan

MAPE	Tingkat Akurasi Peramalan
$\leq 10\%$	<i>Highly accurate</i>
11% - 20%	<i>Good forecast</i>
21% - 50%	<i>Reasonable forecast</i>
51% - lebih	<i>Inaccurate forecast</i>

BAB III

METODELOGI PENELITIAN

3.1 Studi literatur

Pada tahap pertama ini dilakukan identifikasi masalah dan pengkajian tentang pasar *forex*, peramalan dan metode pembelajaran mesin SVR yang dioptimasi dengan algoritma PSO. Pengkajian ini dilakukan dengan membaca jurnal, buku, maupun artikel. Selain itu pada tahap ini juga melakukan diskusi dengan dosen, teman-teman, dan orang-orang yang berkompeten dalam tugas akhir ini.

3.2 Pengumpulan data peramalan

Data yang digunakan dalam penulisan ilmiah ini adalah data *time series* dari kurs EUR/USD pasar *forex online*. Data diambil setiap satu jam dengan bantuan *software trading* Metatrader 4. Penulisan ilmiah ini menggunakan data sebanyak 96 data yang diambil pada tanggal 12 Januari 2014 sampai 16 Januari 2014.

3.3 Variabel Penelitian

Variabel penelitian yang digunakan dalam penelitian adalah data *time series* pergerakan harga kurs EUR/USD pada pasar *forex online* dengan jumlah data sebanyak 96 data. Variabel bebas dan tak bebas yang digunakan adalah sebagai berikut.

1. Variabel tak bebas (y_t) adalah data harga kurs EUR/USD yang dijadikan sebagai target.
2. Variabel bebas (x) adalah data kurs EUR/USD pada waktu sebelumnya ($y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-n}$).

Ada tiga tahapan dalam peramalan harga kurs EUR/USD pada pasar *forex online* yaitu pemodelan untuk mendapatkan nilai bobot (w) dan bias (b) berdasarkan data *training*, memvalidasi model dengan menggunakan data *testing*, setelah itu menghitung

nilai ramalan dengan menggunakan model pada proses *testing*. Model peramalan berdasarkan data *training* adalah :

$$y_{t-1} = f(y_{t-2}, y_{t-3}, y_{t-4}, y_{t-5})$$

Model peramalan berdasarkan data *testing* adalah :

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4})$$

3.4 Peramalan dengan Metode SVR

Berikut ini tahapan-tahapan untuk mendapatkan hasil peramalan dari data harga kurs EUR/USD :

1. Unduh data harga kurs EUR/USD dari *software* Metatrader lalu letakkan data tersebut ke dalam excel. Bagi data menjadi 80% data *training* dan 20% data *testing*.
2. Normalisasi data harga kurs EUR/USD agar variansi tidak terlalu melebar dengan menggunakan persamaan (8)
3. Masukkan data kedalam *compiler* Matlab, bagi data menjadi data *training* dan *testing* kemudian susun data seperti berikut

Tabel 3.1. Penyusunan data

x_1	x_2	x_3	x_4	<i>target</i>
$data_1$	$data_2$	$data_3$	$data_4$	$data_5$
$data_2$	$data_3$	$data_4$	$data_5$	$data_6$
.
.
.
$data_{92}$	$data_{93}$	$data_{94}$	$data_{95}$	$data_{96}$

4. Memetakan data ke *feature space* dengan membentuk matriks kernel dengan fungsi kernel RBF dimana nilai σ sebagai parameternya
5. Hitung optimasi variabel lagrange dengan formulasi *quadratic programming* pada persamaan (2.6) untuk mendapatkan w dan b .

6. Validasi nilai w dan b yang didapat dengan melakukan peramalan menggunakan data *testing*

Langkah-langkah peramalan dengan metode SVR digambarkan dalam bentuk diagram alir yang dapat dilihat pada Gambar 3.1.

3.5 Peramalan dengan Metode PSO-SVR

Berikut ini tahapan-tahapan untuk mendapatkan hasil peramalan dari data harga kurs EUR/USD :

1. Unduh data harga kurs EUR/USD dari *software* Metatrader lalu letakkan data tersebut ke dalam excel. Bagi data menjadi 80% data *training* dan 20% data *testing*.
2. Normalisasi data harga kurs EUR/USD agar variansi tidak terlalu melebar dengan menggunakan persamaan (8)
3. Masukkan data kedalam *compiler* Matlab, bagi data menjadi data *training* dan *testing* kemudian susun data seperti berikut

Tabel 3.1. Penyusunan data

x_1	x_2	x_3	x_4	<i>target</i>
<i>data</i> ₁	<i>data</i> ₂	<i>data</i> ₃	<i>data</i> ₄	<i>data</i> ₅
<i>data</i> ₂	<i>data</i> ₃	<i>data</i> ₄	<i>data</i> ₅	<i>data</i> ₆
.
.
.
<i>data</i> ₉₂	<i>data</i> ₉₃	<i>data</i> ₉₄	<i>data</i> ₉₅	<i>data</i> ₉₆

4. Inisialisasi parameter PSO yaitu parameter C (*penalty cost*)
5. Menentukan posisi (*position*) dari partikel dan kecepatan (*velocity*) dari partikel.

6. Melakukan peramalan menggunakan metode SVR dengan parameter SVR yang dioptimasi menggunakan PSO.
7. Hitung fungsi objektif dari proses *training* yaitu :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - F_i}{X_i} \right| \times 100\%$$

Perhitungan fungsi objektif digunakan untuk mendapatkan nilai kelayakan (*current fitness*) dari tiap-tiap kandidat solusi atau partikel

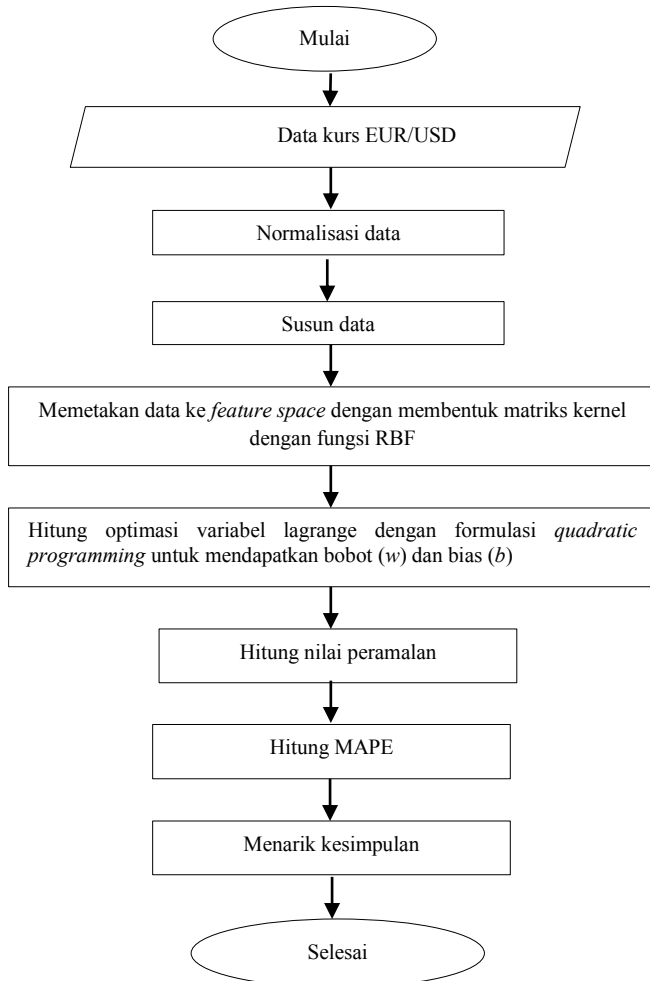
8. Perhitungan fungsi objektif, dapat diketahui *local best fitness* dan *local best position*.
9. Mencari nilai *global best fitness*, yaitu nilai yang minimum dari *local best fitness*.
10. Menentukan *global best position*, yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* dari partikel yang memenuhi persyaratan *global best fitness*.
11. Memperbarui kecepatan dan posisi partikel.
12. Ulangi langkah 5 sampai 11 hingga fungsi objektif konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi.
13. Gunakan bobot (w) dan bias (b) yang telah dioptimasi untuk proses *testing*

Langkah-langkah peramalan dengan metode PSO-SVR digambarkan dalam bentuk diagram alir yang dapat dilihat pada Gambar 3.2.

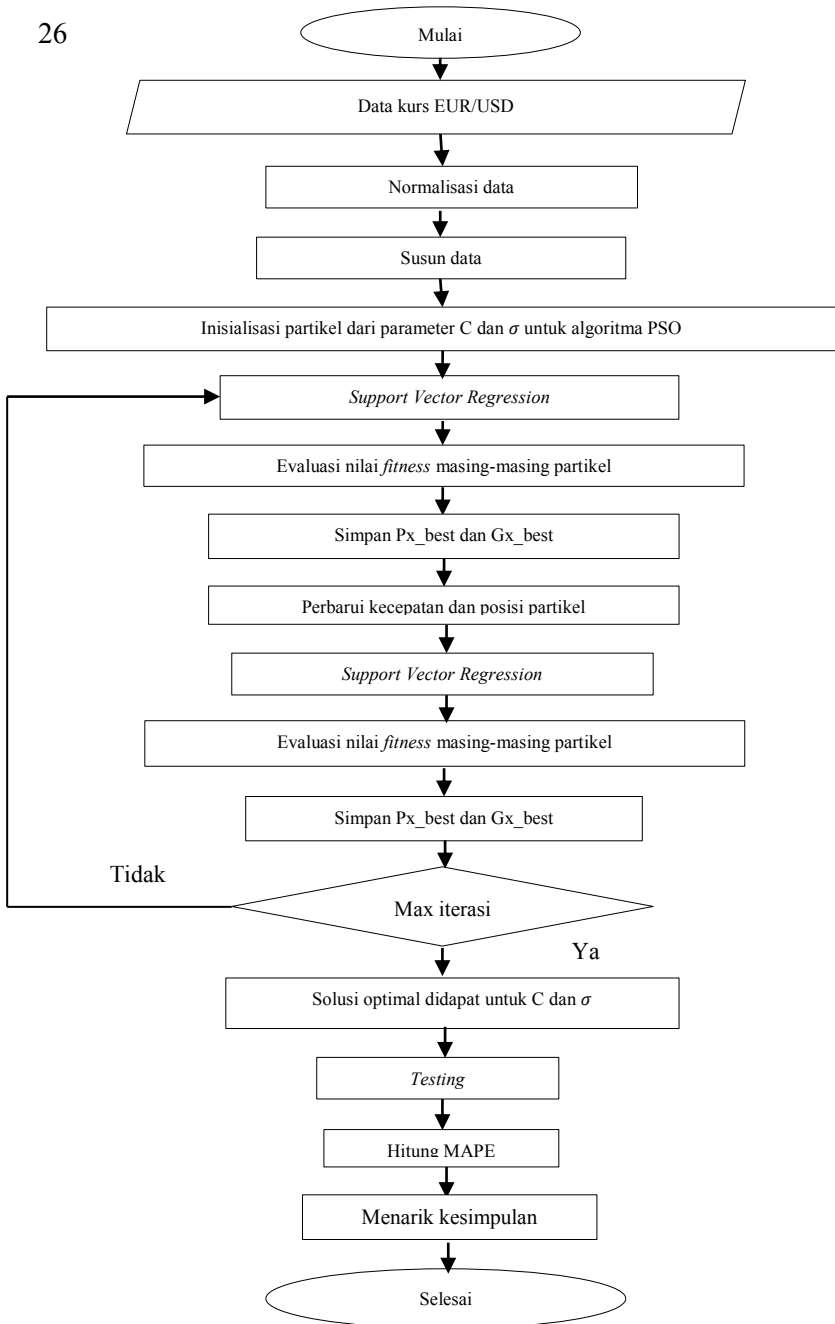
3.6 Analisis Pengolahan Data dan Penarikan Kesimpulan

Dari pengolahan data peramalan dengan menggunakan metode PSO - SVR akan dilakukan analisis hasil pengolahan data. Analisis dan pembahasan yang dilakukan pada penelitian ini adalah menghitung nilai keakuratan model peramalan data *training* menggunakan metode PSO-SVR. Nilai keakuratan dapat dilihat dengan menghitung eror dari peramalan data dengan menggunakan (*Mean Absolute Percentage Error*). Dari nilai

MAPE dapat kita tarik kesimpulan apakah model yang kita dapat dalam proses *training* dengan metode PSO-SVR baik atau buruk.



Gambar 3.1 Diagram alir metode SVR



Gambar 3.2 Diagram Alir Algoritma PSO-SVR

BAB IV ANALISIS DAN PEMBAHASAN

4.1 Data Observasi

Pada penelitian tugas akhir ini data yang digunakan adalah data sekunder kurs *closing* EUR/USD pasar *forex online* Data diambil setiap satu jam dengan bantuan *software trading* Metatrader 4. Data historis diambil selama lima hari yang dimulai pada tanggal 12 Januari 2014 sampai 16 Januari 2014 sehingga didapatkan data sebanyak 120 data. Data kemudian diolah dengan *software* Matlab, menggunakan metode SVR yang dioptimasi dengan algoritma PSO. Data sebanyak 120 data dibagi menjadi 93 data *training* dan 27 data *testing*. Data tersebut dapat dilihat pada Lampiran 1.

4.2 Prapemrosesan Data

Sebelum data diolah menggunakan metode PSO-SVR, data kita proses terlebih dahulu sehingga dapat maksimal saat digunakan. Berikut ini tahapan-tahapan untuk tahapan prapemrosesan data harga kurs EUR/USD :

4.2.1 Pembagian Data

Masukkan data kedalam Matlab, bagi data menjadi data *training*, dan data *testing*, dimana proporsi untuk data *training* yaitu 80% = 93 data dimulai dari data ke-1 hingga data ke-93 dan data *testing* 20% = 27 data dimulai dari data ke-94 hingga data ke-120. Berikut ini *source code* untuk *input* data dari excel ke dalam matlab.

```
%Input Data Kurs Euro terhadap Dollar
dataset = xlsread('data1', 'sheet1');
```

SVR bekerja lebih efektif dengan data yang mempunyai rentang skala kecil. Oleh karena itu data peramalan harga *closing* kurs EUR/USD di normalisasi dengan rentang nilai antara 0 sampai 1. Data dinormalisasi menggunakan metode normalisasi

min-max. Data yang telah ternormalisasi dapat dilihat pada Lampiran 2.

```
function dataskala=normalisasi(dataset);
mindata = min(dataset);
maxdata = max(dataset);
minskala = 0;
maxskala = 1;
% global normal;
dataskala = ((dataset-mindata)/(maxdata-
mindata))*(maxskala-minskala)+minskala;
```

4.2.3 Penyusunan Data

Untuk melakukan peramalan data disusun seperti pola berikut :

	f_1	f_2	f_3	f_4	<i>target</i>
<i>Training</i>	$data_1$	$data_2$	$data_3$	$data_4$	$data_5$
	$data_2$	$data_3$	$data_4$	$data_5$	$data_6$

	$data_{k-4}$	$data_{k-3}$	$data_{k-2}$	$data_{k-1}$	$data_k$
<i>Testing</i>	$data_{k-3}$	$data_{k-2}$	$data_{k-1}$	$data_k$	$data_{k+1}$
	$data_{k-2}$	$data_{k-1}$	$data_k$	$data_{k+1}$	$data_{k+2}$

	$data_{n-4}$	$data_{n-3}$	$data_{n-2}$	$data_{n-1}$	$data_n$

Source code yang digunakan dalam penyusunan data di Matlab adalah sebagai berikut

```
% susuntabel data
for i=1:116
    for j=1:5
        susun(i,j)=dataskala(i+j-1);
    end
end
```

Proses peramalan menggunakan metode SVR terbagi menjadi dua tahap yaitu, proses *training* dan *testing*. Pembentukan model dilakukan pada saat proses *training* kemudian model yang didapat pada proses *training* kita validasi di tahap *testing* menggunakan data *testing*.

Pada penulisan tugas akhir ini, fungsi kernel yang digunakan adalah fungsi kernel RBF. Pada fungsi kernel RBF terdapat suatu parameter σ yang di *inputkan* secara percobaan (*trial and error*).

Fungsi dari kernel RBF adalah sebagai berikut :

$$\text{Kernel RBF} := \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Berikut ini contoh perhitungan fungsi kernel RBF pada indeks $i = 1, j = 1$ dengan menggunakan salah satu nilai partikel dari salah satu nilai pada parameter σ RBF.

$$K(x_1, x_1) = \exp\left(-\frac{\|[0.9959 \ 1 \ 0.9972 \ 0.9698] - [0.9959 \ 1 \ 0.9972 \ 0.9698]\|^2}{2(1.0844)^2}\right)$$

Setelah didapat nilai $K(x_1, x_1)$, selanjutnya nilai tersebut dibawa ke matriks dengan indeks $i = 1$ dan $j = 1$. Nilai indeks i dan j terus berjalan hingga matriks kernel terisi nilai dengan ukuran matriks yang dibentuk sebesar 93×93 .

$$\begin{bmatrix} K(x_1, x_1) & \dots & \dots & K(x_1, x_{93}) \\ \vdots & & & \vdots \\ K(x_1, x_{93}) & \dots & \dots & K(x_{93}, x_{93}) \end{bmatrix}$$

Pembuatan dalam Matlab menggunakan *source code* sebagai berikut :

```

% pembuatan matriks kernel
H = zeros(n,n);
for i=1:n
    for j=1:n
        H(i,j) =
svkernel(ker,X(i,:),X(j,:),p1);
    end
end

```

Dari matriks kernel didapatkan nilai mapping yang selanjutnya dimasukkan ke dalam *quadratic programming* yang persamaannya dapat dilihat pada Persamaan (2.6). Dari penyelesaian *quadratic programming* akan menghasilkan solusi optimal dari *lagrange multiplier* α_i dan α_i^* . Menghitung nilai bobot dan bias menggunakan perintah sebagai berikut. Nilai bobot dan

```

X=susun(1:93,1:4);
Y=susun(1:93,5);
ker='rbf';
loss='quadratic';
e=1;
C=      110      ;
p1=      1.5      ;
%Peramalan menggunakan SVR
[nsv, beta, bias] = svr(X,Y,ker,C,loss,e,p1);

```

Pada metode SVR digunakan variasi *trial and error* parameter SVR sebanyak 3 kali. Nilai bobot yang digunakan untuk Persamaan (4.1) dapat dilihat pada Lampiran 3.

$$f(x) = \sum_{i,j=1}^{93} (\alpha_i - \alpha_i^*) \left(\exp \left(- \frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + b \quad (4.1)$$

Sehingga model yang dihasilkan berdasarkan variasi *trial and eror* parameter sebanyak 3 kali adalah sebagai berikut.

Untuk $C = 190$ dan $\sigma = 1.8$ dihasilkan model sebagai berikut :

$$f(x) = \left((2,8696) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (11,4882) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \right. \\ \left. + (145,1925) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) \right) + 0$$

Untuk $C = 150$ dan $\sigma = 1.4$ dihasilkan model sebagai berikut :

$$f(x) = (2,5413) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (9,3340) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (114,9677) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 110$ dan $\sigma = 1.5$ dihasilkan model sebagai berikut :

$$f(x) = (2,1979) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (7,1699) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (84,6638) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 190$ dan $\sigma = 1.4$ dihasilkan model sebagai berikut :

$$f(x) = (2,1323) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (10,5184) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (143,3856) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 150$ dan $\sigma = 1.5$ dihasilkan model sebagai berikut :

$$f(x) = (1,8890) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (8,4863) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (113,451) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 110$ dan $\sigma = 1.8$ dihasilkan model sebagai berikut :

$$f(x) = (1,6094) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (6,4308) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (83,462) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 190$ dan $\sigma = 1.5$ dihasilkan model sebagai berikut :

$$f(x) = (2,0744) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (10,012) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (135,9062) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 150$ dan $\sigma = 1.8$ dihasilkan model sebagai berikut :

$$f(x) = (1,929) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (8,9962) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (120,9391) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 150$ dan $\sigma = 1.8$ dihasilkan model sebagai berikut :

$$f(x) = (1.929) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (8.9962) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (120.9391) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk $C = 110$ dan $\sigma = 1.4$ dihasilkan model sebagai berikut :

$$f(x) = (4.4865) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (5.11) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (90.9658) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Dengan $x_i =$ data *training* dan $x_j =$ data *testing*.

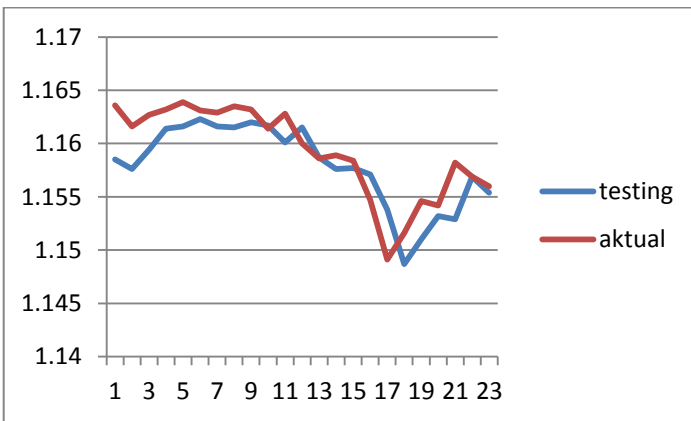
Setelah didapat model terbentuk, dilakukan validasi pada model untuk melihat seberapa baik model yang terbentuk. Validasi dilakukan dengan menggunakan data *testing*. Pada tabel 4.1 adalah nilai keakuratan model yang di validasi dengan menggunakan data *testing*. Proses *testing* dalam Matlab dijalankan dengan menggunakan program berikut :

```
trnX = susun(1:93,1:4);
tstX = susun(94:116,1:4);
tstY= svroutput(trnX,tstX,ker,beta,bias)
```

Tabel 4.1 Keakuratan model SVR

C	σ	MAPE
190	1.8	0.2213%
	1.4	0.2115%
	1.5	0.2090%
150	1.8	0.2080%
	1.4	0.2177%
	1.5	0.2090%
110	1.8	0.2055%
	1.4	0.2180%
	1.5	0.2135%

Dari 9 kali percobaan dapat diketahui SVR menghasilkan eror yang minimum ketika parameter $C = 110$ dan $\sigma = 1.8$ yaitu nilai MAPE = 0.2055%. Data ramalan dan data aktual di denormalisasi terlebih dahulu sebelum dihitung MAPEnya. Plot hasil *testing* data data aktual dapat dilihat pada Gambar 4.1



Gambar 4.1 Perbandingan data *testing* dan aktual

Dari Gambar 4.1 plot hasil *testing* memiliki kemiripan dengan plot data aktual dan nilai MAPE yang didapat pada proses *testing* juga kecil. Maka dapat ditarik kesimpulan model SVR yang didapat pada proses *training* memiliki keakuratan yang baik saat di uji dengan menggunakan data *testing* dengan *inputan* parameter C dan σ yang dilakukan secara *trial and eror*. Pengambilan *inputan* secara *trial and eror* menghasilkan model peramalan yang belum tentu optimal. Oleh karena itu parameter C dan σ sebaiknya di optimasi.

4.4 Peramalan Data Menggunakan Metode PSO-SVR

Proses peramalan menggunakan metode PSO-SVR terbagi menjadi dua tahap yaitu, proses *training* dan *testing*. Pembentukan model dilakukan pada saat proses *training*

kemudian model yang didapat pada proses *training* kita validasi di tahap *testing* menggunakan data *testing*.

Perbedaannya dengan metode SVR adalah parameter tidak kita ambil secara *trial and error* karena parameter SVR akan dioptimasi dengan algoritma PSO. PSO akan membantu SVR untuk mendapatkan fungsi tujuan yaitu MAPE yang kecil.

4.4.1 *Training Data*

Training data merupakan proses untuk mendapatkan nilai w dan b . Nilai dari w dan b akan dimasukkan ke model regresi pada Persamaan (2.1). Untuk mendapatkan nilai w dan b yang baik, maka parameter C (*penalty*) dan σ (kernel RBF) dari metode SVR dioptimasi menggunakan algoritma PSO. Tahapan *training* dalam penelitian ini adalah sebagai berikut :

1. Menentukan ukuran atau jumlah partikel (N) dari parameter PSO yaitu parameter C (*penalty cost*) dan σ (RBF). Untuk mengurangi jumlah evaluasi fungsi yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar, tetapi juga tidak terlalu kecil, agar ada banyak kemungkinan posisi menuju solusi terbaik atau optimal. Jika terlalu kecil, sedikit kemungkinan menemukan posisi partikel yang baik. Terlalu besar juga akan membuat perhitungan jadi panjang. Biasanya digunakan ukuran kawanan adalah 20 sampai 30 partikel. Pada penelitian ini jumlah partikel $N = 20$, kemudian kita merandom nilai awal dengan nilai range C ditetapkan antara 100-200 dan range parameter σ ditetapkan antara 1-2. Berikut ini *source code* dalam penentuan jumlah partikel dari parameter PSO.

```
a1=1; b1=2;
p1=a1+(b1-a1)*rand(n,1);
a2=100; b2=200;
C=a2+(b2-a2)*rand(n,1);
x=[C p1];
```

Tabel 4.2 . Nilai awal parameter

Partikel	C	σ
1	167.8735	1.6324
2	175.7740	1.0975
3	174.3132	1.2785
4	139.2227	1.5469
5	165.5478	1.9575
6	117.1187	1.9649
7	170.6046	1.1576
8	103.1833	1.9706
9	127.6923	1.9572
10	104.6171	1.4854
11	109.7132	1.8003
12	182.3458	1.1419
13	169.4829	1.4218
14	131.7099	1.9157
15	195.0222	1.7922
16	103.4446	1.9595
17	143.8744	1.6557
18	138.1558	1.0357
19	176.5517	1.8491
20	179.5200	1.9340

2. Evaluasi nilai fungsi tujuan untuk setiap partikel. Fungsi tujuan didapat dengan melakukan *testing* atau peramalan menggunakan metode *Support Vector Regression* (SVR). *Testing* atau peramalan pada metode PSO-SVR menggunakan data *training* bukan data *testing*, kemudian kita cari MAPE. Nilai MAPE dijadikan sebagai fungsi tujuan dari PSO. Setelah didapat nilai bobot (w) dan bias (b) Lampiran 4, *testing* nilai tersebut dengan menggunakan data *training* sebagai data *testing*. Dari proses ini didapatkan nilai ramalan kemudian kita hitung nilai MAPE dan dijadikan sebagai fungsi tujuan atau fungsi objektif dari algoritma PSO. MAPE dihitung dengan menggunakan Persamaan (2.12). Pada Tabel 4.3 ditunjukkan nilai dari fungsi objektif

untuk setiap partikel pada iterasi 0. Berikut ini *source code* dalam Matlab untuk menghitung *px_best*, *pf_best*, *gx_best*, *gf_best*.

```
function
[mape,px_best,pf_best,gx_best,gf_best]=eval
l(Y,tstY,x,n);
for i=1:n
a=(Y-tstY(:,i));
bx=abs(a/Y);
dx=bx(1:93,4);
%eror dibagi dengan target
mape(i,1)= (sum(dx)./93)*100;
end
pf_best=mape;
px_best=[x mape];
gx_best=sortrows(px_best,3);
gx_best=gx_best(1,:);
gf_best=gx_best(1,3);
```

Tabel 4.3 Nilai kelayakan (*current fitness*) iterasi 0

Partikel	<i>current fitness</i>	Partikel	<i>current fitness</i>	Partikel	<i>current fitness</i>
1	3.5084	8	3.4675	15	3.5212
2	3.5124	9	3.4844	16	3.4677
3	3.5117	10	3.4686	17	3.4939
4	3.4909	11	3.4725	18	3.4903
5	3.5071	12	3.5156	19	3.5128
6	3.4777	13	3.5092	20	3.5143
7	3.5098	14	3.4868		

- Perhitungan fungsi objektif, dapat diketahui *local best fitness* (*Pf_best*) dan *local best position* (*Px_best*).

Tabel 4.4 Nilai Px_best C, Px_best σ dan Pf_best

Partikel	Px_best C	Px_best σ	Pf_best C dan σ
1	167.8735	1.6324	3.5084
2	175.7740	1.0975	3.5124
3	174.3132	1.2785	3.5117
4	139.2227	1.5469	3.4909
5	165.5478	1.9575	3.5071
6	117.1187	1.9649	3.4777
7	170.6046	1.1576	3.5098
8	103.1833	1.9706	3.4675
9	127.6923	1.9572	3.4844
10	104.6171	1.4854	3.4686
11	109.7132	1.8003	3.4725
12	182.3458	1.1419	3.5156
13	169.4829	1.4218	3.5092
14	131.7099	1.9157	3.4868
15	195.0222	1.7922	3.5212
16	103.4446	1.9595	3.4677
17	143.8744	1.6557	3.4939
18	138.1558	1.0357	3.4903
19	176.5517	1.8491	3.5128
20	179.5200	1.9340	3.5143

4. Mencari nilai *global best fitness* (Gf_best), yaitu nilai yang minimum dari *local best fitness* (Pf_best). Dari perhitungan didapat nilai *global best fitness* (Gf_best) sebesar 3.4675 pada partikel 8.
5. Menentukan *global best position* (Gx_best), yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* (Px_best). dari partikel yang memenuhi persyaratan *global best fitness* (Gf_best). Sehingga didapatkan partikel yang menghasilkan nilai *global best fitness* (Gf_best) sebesar 3.4675 adalah pada saat $C = 103.1833$ dan $\sigma = 1.9706$.

6. Memperbarui kecepatan dengan menggunakan persamaan (2.8). Kecepatan partikel pada iterasi 1 dapat dilihat pada tabel 4.5.

Tabel 4.5 Kecepatan partikel pada iterasi pertama

j	$V_j(1)C$	$V_j(1) \sigma$
1	-63.3659	0.3313
2	-71.1047	0.8552
3	-69.6738	0.6779
4	-35.3016	0.4150
5	-61.0878	0.0128
6	-13.6501	0.0056
7	-66.0411	0.7963
8	0	0
9	-24.0073	0.0132
10	-1.4045	0.4753
11	-6.3962	0.1668
12	-77.5419	0.8117
13	-64.9423	0.5376
14	-27.9427	0.0537
15	-89.9589	0.1747
16	-0.2560	0.0109
17	-39.8582	0.3084
18	-34.2566	0.9157
19	-71.8665	0.1190
20	-74.7740	0.0359

7. Memperbarui posisi partikel dengan Persamaan (2.9). Sehingga didapat posisi terbaru pada tabel

Tabel 4.6 Posisi partikel yang diperbarui

j	$X_j(1) C$	$X_j(1) \sigma$
1	104.5076	1.9637
2	104.6693	1.9527
3	104.6394	1.9564
4	103.9211	1.9619
5	104.4600	1.9703
6	103.4686	1.9705
7	104.5635	1.9540

j	$X_j(1) C$	$X_j(1) \sigma$
8	103.1833	1.9706
9	103.6850	1.9703
10	103.2126	1.9607
11	103.3170	1.9671
12	104.8038	1.9536
13	104.5405	1.9594
14	103.7673	1.9695
15	105.0633	1.9669
16	103.1886	1.9704
17	104.0163	1.9641
18	103.8992	1.9515
19	104.6852	1.9681
20	104.7460	1.9698

Kecepatan dan posisi partikel diperbarui dengan menggunakan *source code* berikut ini.

```
function [x,v]=update(px_best,gx_best,x,n)
c1=2; c2=2;
r1=rand; r2=rand;
v(n,1)=[0];
    v(n,2)=[0];
for i=1:n
v(i,:)=v(i)+(c1*r1).*(px_best(i,1:2)-
x(i,:))+(c2*r2).*(gx_best(1,1:2)-x(i,:));
end
x=x+v;
    return
```

- Setelah didapat posisi partikel yang baru, hitung kembali fungsi tujuan dengan menggunakan posisi partikel sekarang. Jika nilai Pf_best sekarang lebih baik dari nilai Pf_best dari iterasi sebelumnya, maka nilai Px_best sebelumnya digantikan dengan Px_best sekarang. Berikut ini *source code* untuk langkah ke-8.

```

for i=1:n
  if px_bestit(i,3)<px_best(i,3)
    px_best(i,:)=px_bestit(i,:);
    x(i,:)=x_it(i,:);
  else
    px_best(i,:)=px_best(i,:);
    x(i,:)=x(i,:);
end

```

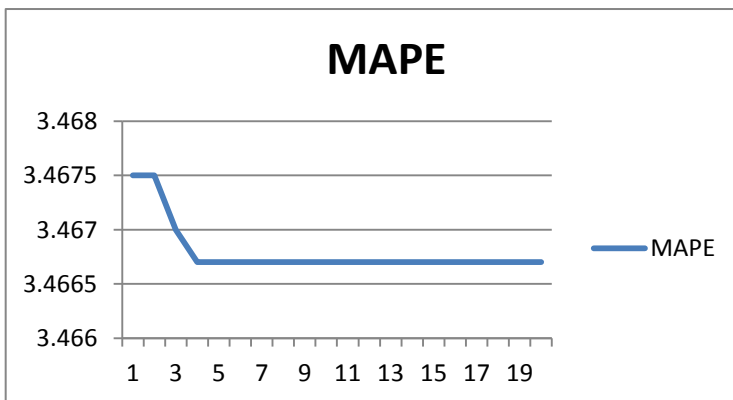
9. Ulangi langkah 3 sampai 8 hingga fungsi objektif atau nilai Gf_best konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi. Pada penelitian ini nilai dari fungsi objektif (MAPE) telah konvergen pada iterasi ke-6 dapat dilihat pada Tabel 4.7. Maka didapatkan nilai masukkan C dan σ terbaik untuk mendapatkan nilai w dan b yang paling baik pula.

Tabel 4.7 Partikel dan MAPE konvergen pada suatu nilai

Partikel	C	σ	MAPE
1	102.2522	1.9724	3.4667
2	102.2522	1.9724	3.4667
3	102.2522	1.9724	3.4667
4	102.2522	1.9724	3.4667
5	102.2522	1.9724	3.4667
6	102.2522	1.9724	3.4667
7	102.2522	1.9724	3.4667
8	102.2522	1.9724	3.4667
9	102.2522	1.9724	3.4667
10	102.2522	1.9724	3.4667
11	102.2522	1.9724	3.4667
12	102.2522	1.9724	3.4667
13	102.2522	1.9724	3.4667

Partikel	C	σ	MAPE
14	102.2522	1.9724	3.4667
15	102.2522	1.9724	3.4667
16	102.2522	1.9724	3.4667
17	102.2522	1.9724	3.4667
18	102.2522	1.9724	3.4667
19	102.2522	1.9724	3.4667
20	102.2522	1.9724	3.4667

Pergerakan nilai MAPE menuju konvergen ke suatu nilai dapat dilihat pada Gambar 4.2 Nilai partikel konvergen kesuatu nilai pada iterasi ke 8. Pada Gambar 4.2 nilai MAPE pada proses *training* bergerak menuju nilai MAPE yang minimum disetiap iterasi. Nilai MAPE mulai konvergen pada iterasi ke-5 dan pada iterasi ke-8, nilai dari partikel C dan σ tidak berubah lagi. Sehingga didapatkan nilai optimal dari $C = 102.2522$ dan $\sigma = 1.9724$.



Gambar 4.2 Pergerakan nilai MAPE pada proses *training* PSO-SVR

4.4.2 *TestingData*

Untuk melihat seberapa baik model yang terbentuk, kita lakukan validasi model dengan menggunakan data *testing*. Data yang digunakan untuk validasi model adalah data pada periode 94 hingga periode 116. Proses *training* mendapatkan nilai bobot (w) dan bias (b) dengan parameter yang paling optimal adalah $C = 102.2522$ dan $\sigma = 1.9724$. Model yang dihasilkan untuk proses *testing* adalah :

$$f(x) = \left((18.280) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (64.017) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \right. \\ \left. + (783.207) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) \right) + 0$$

dengan $x_i =$ data *training* dan $x_j =$ data *testing*.

Dari proses *testing* didapatkan nilai peramalan yang dapat dilihat pada Tabel 4.8. Proses *testing* pada Matlab menggunakan perintah program seperti berikut ini.

```
%testing
trnX = susun(1:93,1:4);
tstA= susun(94:116,1:4);
tstY = svroutput(trnX,tstA,ker,beta,bias);
```

Tabel 4.8. Perbandingan data aktual dan data peramalan

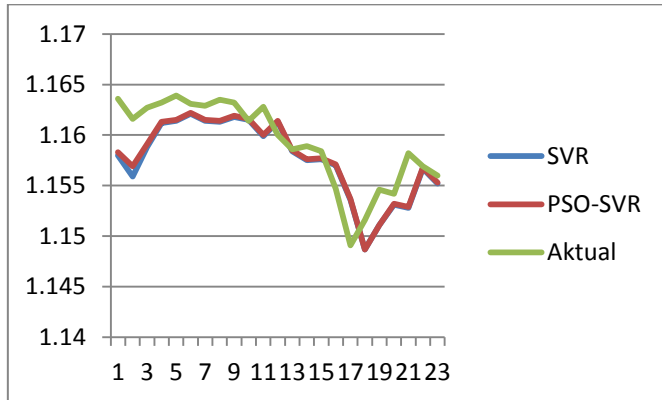
Periode	Aktual	Peramalan
94	1.1636	1.1583
95	1.1616	1.1569
96	1.1627	1.1591
97	1.1632	1.1613
98	1.1639	1.1615
99	1.1631	1.1622
100	1.1629	1.1615

Periode	Aktual	Peramalan
101	1.1635	1.1614
102	1.1632	1.1619
103	1.1614	1.1616
104	1.1628	1.1600
105	1.1600	1.1614
106	1.1586	1.1585
107	1.1589	1.1576
108	1.1584	1.1577
109	1.1547	1.1571
110	1.1491	1.1537
111	1.1516	1.1487
112	1.1546	1.1511
113	1.1542	1.1532
114	1.1582	1.1529
115	1.1569	1.1568
116	1.1560	1.1553

Keakuratan nilai peramalan dihitung dengan menggunakan *mean absolute percentage error* (MAPE). Nilai MAPE yang didapat adalah 0.1895%. Dari perhitungan error dengan menggunakan MAPE dihasilkan nilai yang kecil. Hal ini menunjukkan hasil ramalan memiliki nilai error yang kecil. Perhitungan nilai akurasi data *testing* telah didenormalisasi dalam Matlab dengan *source code* sebagai berikut.

```
%denormalisasi
mindata=min(dataset(:));
maxdata=max(dataset(:));
rentang=maxdata-mindata;
denormalisasi= (tstY*rentang)+ mindata
aktual=datasetsusun(94:116,5)
a=(aktual-denormalisasi);
dx=abs(a/aktual)
fx=dx(1:23,5)
%error dibagi dengan target
mape = (sum(fx)./23)*100
```

Berikut ini Gambar 4.3 ditunjukkan perbandingan pergerakan data aktual dan data peramalan



Gambar 4.3 Plot data *testing* dan data aktual

Dari Gambar 4.3 terlihat bahwa bentuk pergerakan grafik data peramalan menyerupai pergerakan grafik data aktual. Metode PSO-SVR menghasilkan nilai MAPE yang sangat kecil dan lebih kecil dibandingkan peramalan dengan menggunakan metode SVR. Maka dapat ditarik kesimpulan model PSO-SVR yang didapat pada proses *training* memiliki keakuratan yang baik daripada model yang dihasilkan metode SVR saat di uji dengan menggunakan data *testing*. Hal ini juga menunjukkan bahwa algoritma PSO berhasil untuk mengoptimasi parameter dari SVR.

“Halaman ini sengaja dikosongkan”

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan analisis terhadap hasil perhitungan peramalan yang telah dilakukan terhadap data *closing* kurs EUR/USD pada pasar *forex online* dengan menggunakan metode *support vector regression* (SVR) yang dioptimasi dengan algoritma *particle swarm optimization* (PSO) maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Parameter dari SVR yang telah dioptimasi dengan menggunakan PSO optimal pada saat C bernilai 102.2522 dan σ bernilai 1.9724.
2. Tugas Akhir ini telah berhasil melakukan peramalan data kurs EUR/USD pada pasar *forex online* menggunakan metode peramalan SVR dengan nilai parameter C = 110 dan $\sigma = 1.8$ didapatkan MAPE sebesar 0.2055%. Untuk hasil peramalan menggunakan SVR dengan parameter yang telah optimal C = 102.2522 dan $\sigma = 1.9724$ didapatkan MAPE sebesar 0.1895%.

5.2 Saran

Dari Penelitian yang telah dilakukan, maka disarankan untuk mencoba menggunakan jenis data (studi kasus) dan algoritma optimasi parameter yang lain dalam menghitung nilai peramalan menggunakan metode SVR.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- [1] Hadi, H. 1997. “Valas Untuk Manajer (Forex for Managers)”. Jakarta. Ghalia Indonesia.
- [2] Anonim. 2009. “Struktur Investasi”. <http://belajarforex.com/sitting-duck/1-struktur-nvestasi.html>. Diakses pada tanggal 6 Februari 2015.
- [3] Kamruzzaman, J. dan Ruhul A Sarker. 2004. “Aplication of Support Vector Machine to Forex Monitoring”. IEEJ.
- [4] Tomasouw, B. P. dan Mohammad Isa Irawan. 2012. “*Multiclass Twin Bounded Support Vector Machine* Untuk Pengenalan Ucapan”. Tesis. Jurusan Matematika ITS. Surabaya.
- [5] Supriyanto, H. 2012. “Implementasi Support Vector Machines Untuk Memprediksi Arah Pergerakan Harga Harian Valuta Asing (EUR/USD, GBP/USD, dan USD/JPY) Dengan Metode Kernel Trick Menggunakan Fungsi Kernel Radial Basis Function”. Tugas Akhir. Jurusan Matematika Universitas Brawijaya. Malang.
- [6] Wang, X, Junhao Wen, Yihao Zhang, Yubiao Wang. 2013. “Real Estate Price Forecasting Based on SVM Optimized by PSO”. *Optik*, 125 (2014) 1439-1443.
- [7] J. Kennedy and R.C. Eberhart.1995. “A New Optimizer Using Particle Swarm Theory”.
- [8] Amiri, M., Zandieh, M., Vahdani, B., Soltani, R., & Roshanaei, V. 2010. “An integrated eigenvector –DEA – TOPSIS methodology for portfolio risk evaluation in the FOREX spot market”. *Journal of Expert Systems with Applications*.
- [9] Investopedia Staff. “Forex Tutorial: Forex History and Market Participants”. <http://www.investopedia.com/university/forexmarket/forex4.asp>. diakses tanggal 6 Februari 2015.
- [10] Nasution, A. H. 1999. “Metode Aplikasi Peramalan (2 ed, Vol. 1)”. Binarupa Aksara

- [11] Makridakis, S., & Wheelwright, S. C. 1999. "Forecasting Methods for Management". New York: John Wiley & Sons.
- [12] Santosa, B. 2007. "Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta: Graha Ilmu.
- [13] B. Scholkopf and A.J. Smola. 2002. "Learning with Kernels". The MIT Press, Cambridge, Massachusetts.
- [14] Santosa, B. "Tutorial Particle Swarm Optimization". Teknik Industri ITS. Surabaya.
- [15] Lawrence, D. Kenneth. dkk. 2009. "*Fundamentals of Forecasting Using Excel*". Industrial Press, Inc. New York.

Lampiran 1**Data Time Series Closing Price kurs EUR/USD**

	12-Jan	13-Jan	14-Jan	15-Jan	16-Jan
0:00	1.1866	1.1829	1.1776	1.179	1.1627
1:00	1.1869	1.1832	1.1777	1.1784	1.1636
2:00	1.1867	1.1842	1.1771	1.1776	1.1616
3:00	1.1857	1.1844	1.1774	1.1775	1.1627
4:00	1.1847	1.1845	1.1779	1.178	1.1632
5:00	1.1855	1.1846	1.1774	1.1781	1.1639
6:00	1.1853	1.1836	1.1786	1.1774	1.1631
7:00	1.1861	1.1828	1.1792	1.1767	1.1629
8:00	1.1856	1.183	1.1785	1.1772	1.1635
9:00	1.1854	1.185	1.18	1.1777	1.1632
10:00	1.1815	1.1821	1.1792	1.1759	1.1614
11:00	1.1813	1.1808	1.1744	1.1723	1.1628
12:00	1.1795	1.1802	1.1759	1.1686	1.16
13:00	1.1799	1.1793	1.1754	1.1693	1.1586
14:00	1.179	1.1781	1.1757	1.1737	1.1589
15:00	1.1824	1.177	1.1825	1.1699	1.1584
16:00	1.184	1.1762	1.179	1.1631	1.1547
17:00	1.1817	1.1781	1.1786	1.1586	1.1491
18:00	1.1831	1.1778	1.1806	1.1596	1.1516
19:00	1.1826	1.1784	1.1788	1.1601	1.1546
20:00	1.1841	1.1767	1.1785	1.1616	1.1542
21:00	1.1839	1.1764	1.1776	1.1613	1.1582
22:00	1.1837	1.1771	1.1789	1.1618	1.1569
23:00	1.1834	1.1773	1.1789	1.163	1.156

Lampiran 2
Data Ternormalisasi

NO	Periode	Harga	Bentuk Normal
1	0:00	1.1866	0.995879
2	1:00	1.1869	1
3	2:00	1.1867	0.997253
4	3:00	1.1857	0.983516
5	4:00	1.1847	0.96978
6	5:00	1.1855	0.980769
7	6:00	1.1853	0.978022
8	7:00	1.1861	0.989011
9	8:00	1.1856	0.982143
10	9:00	1.1854	0.979396
11	10:00	1.1815	0.925824
12	11:00	1.1813	0.923077
13	12:00	1.1795	0.898352
14	13:00	1.1799	0.903846
15	14:00	1.1790	0.891484
16	15:00	1.1824	0.938187
17	16:00	1.1840	0.960165
18	17:00	1.1817	0.928571
19	18:00	1.1831	0.947802
20	19:00	1.1826	0.940934
21	20:00	1.1841	0.961538
22	21:00	1.1839	0.958791
23	22:00	1.1837	0.956044
24	23:00	1.1834	0.951923
25	0:00	1.1564	0.581044

Lampiran 2 (Lanjutan)
Data Ternormalisasi

NO	Periode	Harga	Bentuk Normal
26	1:00	1.1560	0.575549
27	2:00	1.1561	0.576923
28	3:00	1.1570	0.589286
29	4:00	1.1556	0.570055
30	5:00	1.1560	0.575549
31	6:00	1.1556	0.570055
32	7:00	1.1561	0.576923
33	8:00	1.1568	0.586538
34	9:00	1.1557	0.571429
35	10:00	1.1583	0.607143
36	11:00	1.1615	0.651099
37	12:00	1.1600	0.630495
38	13:00	1.1587	0.612637
39	14:00	1.1605	0.637363
40	15:00	1.1610	0.644231
41	16:00	1.1621	0.659341
42	17:00	1.1630	0.671703
43	18:00	1.1621	0.659341
44	19:00	1.1608	0.641484
45	20:00	1.1614	0.649725
46	21:00	1.1601	0.631868
47	22:00	1.1608	0.641484
48	23:00	1.1606	0.638736
49	0:00	1.1143	0.002747
50	1:00	1.1141	0

Lampiran 2 (Lanjutan)
Data Ternormalisasi

NO	Periode	Harga	Bentuk Normal
51	2:00	1.1167	0.035714
52	3:00	1.1168	0.037088
53	4:00	1.1153	0.016484
54	5:00	1.1175	0.046703
55	6:00	1.1172	0.042582
56	7:00	1.1184	0.059066
57	8:00	1.1216	0.103022
58	9:00	1.1238	0.133242
59	10:00	1.1236	0.130495
60	11:00	1.1250	0.149725
61	12:00	1.1228	0.119505
62	13:00	1.1224	0.114011
63	14:00	1.1244	0.141484
64	15:00	1.1268	0.174451
65	16:00	1.1271	0.178571
66	17:00	1.1255	0.156593
67	18:00	1.1288	0.201923
68	19:00	1.1279	0.18956
69	20:00	1.1274	0.182692
70	21:00	1.1266	0.171703
71	22:00	1.1262	0.166209
72	23:00	1.1237	0.131868
73	0:00	1.1311	0.233516
74	1:00	1.1307	0.228022
75	2:00	1.1308	0.229396

Lampiran 2 (Lanjutan)
Data Ternormalisasi

NO	Periode	Harga	Bentuk Normal
76	3:00	1.1315	0.239011
77	4:00	1.1318	0.243132
78	5:00	1.1310	0.232143
79	6:00	1.1308	0.229396
80	7:00	1.1308	0.229396
81	8:00	1.1307	0.228022
82	9:00	1.1315	0.239011
83	10:00	1.1322	0.248626
84	11:00	1.1352	0.289835
85	12:00	1.1339	0.271978
86	13:00	1.1328	0.256868
87	14:00	1.1315	0.239011
88	15:00	1.1311	0.233516
89	16:00	1.1341	0.274725
90	17:00	1.1342	0.276099
91	18:00	1.1336	0.267857
92	19:00	1.1332	0.262363
93	20:00	1.1341	0.274725
94	21:00	1.1345	0.28022
95	22:00	1.1335	0.266484
96	23:00	1.1341	0.274725
97	0:00	1.1627	0.667582
98	1:00	1.1636	0.679945
99	2:00	1.1616	0.652473
100	3:00	1.1627	0.667582

Lampiran 2 (Lanjutan)
Data Ternormalisasi

NO	Periode	Harga	Bentuk Normal
101	4:00	1.1632	0.674451
102	5:00	1.1639	0.684066
103	6:00	1.1631	0.673077
104	7:00	1.1629	0.67033
105	8:00	1.1635	0.678571
106	9:00	1.1632	0.674451
107	10:00	1.1614	0.649725
108	11:00	1.1628	0.668956
109	12:00	1.16	0.630495
110	13:00	1.1586	0.611264
111	14:00	1.1589	0.615385
112	15:00	1.1584	0.608516
113	16:00	1.1547	0.557692
114	17:00	1.1491	0.480769
115	18:00	1.1516	0.51511
116	19:00	1.1546	0.556319
117	20:00	1.1542	0.550824
118	21:00	1.1582	0.605769
119	22:00	1.1569	0.587912
120	23:00	1.156	0.575549

Lampiran 3

Bobot dan Bias Metode SVR dengan $C = 190$ dan $\sigma = 1.8$

W1	2.829896	W31	19.95333	W61	-1.1533	W91	-7.956
W2	11.44178	W32	23.34478	.	-12.0971	W92	-0.796
W3	7.16091	.	2.176064	.	10.2465	W93	145.129
W4	13.31844	.	2.600189	.	-9.6598		
W5	6.949831	.	15.72694	.	-6.1097		
W6	8.143378	.	9.827046	.	-11.0521		
W7	-11.7514	.	14.68677	.	-8.78128		
W8	5.302463	.	13.42521	.	-19.9133		
W9	-3.50464	.	4.560159	.	30.3734		
W10	9.165571	.	1.865769	.	-4.8719		
W11	3.477107	.	10.36081	.	1.4219		
W12	26.59297	.	0.555369	.	-1.7524		
W13	19.18513	.	11.44412	.	-2.6366		
W14	1.096095	.	6.171621	.	-8.0527		
W15	17.80337	.	-233.568	.	-5.8991		
W16	6.051782	.	-4.33155	.	-5.4100		
W17	17.19635	.	-5.65492	.	-5.3780		
W18	9.133689	.	-2.38374	.	-0.665		
W19	9.108654	.	-12.1945	.	-0.6767		
W20	7.770803	.	3.03163	.	12.1455		
W21	-132.255	.	-8.186	.	-8.5817		
W22	0.52515	.	0.926384	.	-7.894		
W23	-0.0358	.	10.47168	.	-12.2538		
W24	13.11181	.	8.5300	.	-7.5327		
W25	-0.62728	.	-2.75546	.	10.1325		
W26	8.533817	.	2.590473	.	-2.2804		
W27	3.401019	W57	-17.1068	W87	-4.6392		
W28	8.884413	W58	-9.07109	W88	-6.239		
W29	9.69389	W59	1.5123	W89	0.273		
W30	1.061004	W60	1.51231	W90	-5.89915		

Lampiran 3 (Lanjutan)**Bobot dan bias Metode SVR dengan $C = 150$ dan $\sigma = 1.4$**

W1	2.541258	W31	15.59753	W61	-0.97088	W91	-6.0142
W2	9.334001	W32	18.36194	.	-9.56275	W92	-0.34872
W3	5.961283	.	1.491337	.	8.185626	W93	114.9677
W4	10.73614	.	1.706465	.	-7.40727		
W5	5.71919	.	12.21898	.	-4.80949		
W6	6.65648	.	7.613125	.	-8.51682		
W7	-9.03124	.	11.32628	.	-6.76508		
W8	4.342111	.	10.34676	.	-15.5616		
W9	-2.5625	.	3.310881	.	24.06489		
W10	7.30274	.	1.157396	.	-3.58267		
W11	2.739294	.	7.922935	.	0.980942		
W12	20.90877	.	0.237563	.	-1.10512		
W13	15.11943	.	8.765384	.	-1.82572		
W14	0.761801	.	4.683217	.	-6.1196		
W15	13.92501	.	-184.648	.	-4.40768		
W16	4.888338	.	-4.16988	.	-3.98158		
W17	13.63508	.	-4.58006	.	-3.98572		
W18	7.278917	.	-2.04564	.	-0.27511		
W19	7.251802	.	-10.3467	.	-0.26972		
W20	6.219135	.	1.814438	.	9.826247		
W21	-104.285	.	-6.93157	.	-6.49574		
W22	-0.15221	.	0.125286	.	-6.05948		
W23	-0.04039	.	7.825541	.	-9.32566		
W24	10.44582	.	6.347301	.	-5.63646		
W25	-0.60959	.	-2.51189	.	8.312373		
W26	6.570247	.	1.869534	.	-1.48962		
W27	2.61498	W57	-13.4972	W87	-3.49185		
W28	6.889027	W58	-7.2312	W88	-4.6253		
W29	7.562766	W59	1.248779	W89	0.530943		
W30	0.713614	W60	5.568313	W90	-0.83782		

Lampiran 3 (Lanjutan)

Bobot dan bias Metode SVR dengan $C = 110$ dan $\sigma = 1.5$

W1	2.197936	W31	11.27289	W61	-0.74791	W91	-4.11688
W2	7.169888	W32	13.39126	.	-7.00521	W92	0.03916
W3	4.708646	.	0.886402	.	6.092178	W93	84.66384
W4	8.121553	.	0.902603	.	-5.20431		
W5	4.454383	.	8.732609	.	-3.48417		
W6	5.135655	.	5.424457	.	-6.04366		
W7	-6.35204	.	8.035959	.	-4.79526		
W8	3.347978	.	7.326217	.	-11.2565		
W9	-1.66493	.	2.131824	.	17.71791		
W10	5.427103	.	0.516383	.	-2.34061		
W11	2.005586	.	5.525847	.	0.651017		
W12	15.24906	.	-0.04692	.	-0.52976		
W13	11.07471	.	6.140954	.	-1.06458		
W14	0.479609	.	3.225127	.	-4.22852		
W15	10.08304	.	-135.672	.	-2.97075		
W16	3.708465	.	-3.98379	.	-2.62368		
W17	10.07484	.	-3.42091	.	-2.65013		
W18	5.4235	.	-1.68296	.	0.06171		
W19	5.398312	.	-8.34673	.	0.083785		
W20	4.659887	.	0.682182	.	7.468153		
W21	-76.3316	.	-5.60561	.	-4.44691		
W22	-0.78888	.	-0.54965	.	-4.23625		
W23	-0.0687	.	5.251419	.	-6.49135		
W24	7.714722	.	4.259209	.	-3.82057		
W25	-0.56626	.	-2.16384	.	6.412849		
W26	4.640955	.	1.194798	.	-0.75299		
W27	1.831049	W57	-9.89415	W87	-2.34769		
W28	4.919279	W58	-5.38149	W88	-3.07893		
W29	5.446398	W59	0.940455	W89	0.714975		
W30	0.396437	W60	4.070407	W90	-0.27962		

Lampiran 3 (Lanjutan)**Bobot dan bias Metode SVR dengan $C = 190$ dan $\sigma = 1.4$**

W1	2.1323	W31	20.7400	W61	-0.0533	W91	-9.1988
W2	10.5184	W32	23.7187	.	-11.5664	W92	-2.3605
W3	6.4187	.	3.7554	.	9.4891	W93	143.3856
W4	12.8309	.	4.5725	.	-10.2648		
W5	6.6568	.	16.4818	.	-6.0147		
W6	7.7092	.	10.4248	.	-12.2319		
W7	-12.2850	.	16.1213	.	-9.9238		
W8	4.2034	.	14.7151	.	-20.9650		
W9	-4.7003	.	6.0218	.	29.2373		
W10	8.1863	.	3.3214	.	-4.8913		
W11	3.0216	.	11.3455	.	3.0397		
W12	26.3659	.	1.3119	.	-2.9040		
W13	19.4088	.	12.6295	.	-3.8367		
W14	1.9433	.	6.9278	.	-9.1451		
W15	18.1744	.	-232.3923	.	-7.2846		
W16	5.6335	.	-2.0126	.	-7.0372		
W17	16.8209	.	-4.5741	.	-6.7758		
W18	9.1045	.	-2.0560	.	-1.9910		
W19	8.9988	.	-9.2578	.	-1.9152		
W20	7.5252	.	4.5964	.	11.1338		
W21	-132.7724	.	-6.2184	.	-9.3904		
W22	1.1471	.	3.2383	.	-8.6240		
W23	0.2261	.	12.4193	.	-14.3292		
W24	12.8460	.	11.1052	.	-9.5060		
W25	-0.0125	.	-0.1535	.	8.2637		
W26	9.3800	.	3.7465	.	-3.4584		
W27	3.6458	W57	-16.8174	W87	-5.1460		
W28	9.5127	W58	-9.2028	W88	-7.8190		
W29	10.1334	W59	0.7735	W89	-1.5106		
W30	1.7558	W60	7.3590	W90	-3.0362		

Lampiran 3 (Lanjutan)

Bobot dan bias Metode SVR dengan $C = 150$ dan $\sigma = 1.5$

W1	1.8890922	W31	16.297881	W61	-0.1652735	W91	-7.1363447
W2	8.4863894	W32	18.788679		-9.2334843	W92	-1.7046659
W3	5.2548953	.	2.7906356	.	7.533297	W93	113.451029
W4	10.264302	.	3.3513543	.	-7.9283163		
W5	5.428627	.	12.987383	.	-4.8961611		
W6	6.2378892	.	8.1970994	.	-9.4972656		
W7	-9.5271991	.	12.521735	.	-7.7457698		
W8	3.3787869	.	11.500468	.	-16.45374		
W9	-3.6168175	.	4.5729741	.	23.117398		
W10	6.4233239	.	2.4209895	.	-3.6205964		
W11	2.3539427	.	8.8322321	.	1.9750074		
W12	20.697761	.	0.9516103	.	-2.0673741		
W13	15.31704	.	9.7801372	.	-2.8925487		
W14	1.4486227	.	5.4186829	.	-7.1138033		
W15	14.288843	.	-183.6396	.	-5.6207212		
W16	4.5319663	.	-2.0017215	.	-5.3623024		
W17	13.270012	.	-3.8502317	.	-5.2036755		
W18	7.2530975	.	-1.730476	.	-1.4365171		
W19	7.130015	.	-7.6788272	.	-1.3600444		
W20	6.0078867	.	3.4271702	.	8.8918457		
W21	-104.75369	.	-5.0427394	.	-7.2643699		
W22	0.4826187	.	2.2240503	.	-6.8247589		
W23	0.2598719	.	9.6521671	.	-11.032777		
W24	10.327625	.	8.5478615	.	-7.2935224		
W25	-0.0376864	.	-0.404853	.	6.7430602		
W26	7.3085651	.	2.7998191	.	-2.5338402		
W27	2.9355747	W57	-13.259185	W87	-4.0875296		
W28	7.4475956	W58	-7.3680929	W88	-5.9746211		
W29	8.0111529	W59	0.7138174	W89	-0.9795122		
W30	1.3391398	W60	5.7943627	W90	-2.1915186		

Lampiran 3 (Lanjutan)**Bobot dan bias Metode SVR dengan $C = 110$ dan $\sigma = 1.8$**

W1	1.6094627	W31	11.860707	W61	-0.2226175	W91	-5.0744893
W2	6.4308205	W32	13.813989	.	-6.8492802	W92	-1.0678354
W3	4.0719584	.	1.8780335	.	5.5712097	W93	83.462909
W4	7.6825536	.	2.1845507	.	-5.6347229		
W5	4.1680358	.	9.4533564	.	-3.6928387		
W6	4.7417318	.	5.9599441	.	-6.8055573		
W7	-6.7997575	.	8.9754616	.	-5.5795124		
W8	2.5548846	.	8.290261	.	-11.9613		
W9	-2.5264808	.	3.1564228	.	16.991888		
W10	4.7034599	.	1.5527957	.	-2.4124651		
W11	1.7097867	.	6.323006	.	1.1162319		
W12	15.077957	.	0.592985	.	-1.2937438		
W13	11.235046	.	6.9713898	.	-1.9598668		
W14	0.9838687	.	3.8938615	.	-5.0818263		
W15	10.397629	.	-134.84983	.	-3.9689971		
W16	3.4195679	.	-2.006172	.	-3.7244217		
W17	9.7444005	.	-3.0274362	.	-3.648123		
W18	5.3899141	.	-1.4090315	.	-0.895754		
W19	5.2660736	.	-6.0696617	.	-0.823119		
W20	4.4773113	.	2.2237596	.	6.6536358		
W21	-76.735726	.	-3.9082474	.	-5.1467474		
W22	-0.1465207	.	1.2447189	.	-4.9724679		
W23	0.2708738	.	6.8594192	.	-7.8087164		
W24	7.7444332	.	6.0191021	.	-5.1222977		
W25	-0.0697482	.	-0.5804688	.	5.1777535		
W26	5.25072	.	1.893063	.	-1.6358895		
W27	2.1793363	W57	-9.7123098	W87	-2.969225		
W28	5.3911848	W58	-5.5000591	W88	-4.1676224		
W29	5.8646367	W59	0.6030916	W89	-0.4882232		
W30	0.9236731	W60	4.233964	W90	-1.3797177		

Lampiran 3 (Lanjutan)

Bobot dan bias Metode SVR dengan $C = 190$ dan $\sigma = 1.5$

W1	2.074482	W31	19.62919	W61	-0.085797	W91	-8.68334
W2	10.01211	W32	22.4903		-10.987408	W92	-2.19519
W3	6.129051		3.510097		9.00049775	W93	135.9062
W4	12.19047		4.263119		-9.6772792		
W5	6.352541		15.61185		-5.7422752		
W6	7.343491		9.868702		-11.544865		
W7	-11.5931		15.21699		-9.3786267		
W8	3.997015		13.91145		-19.835942		
W9	-4.43064		5.657126		27.70738		
W10	7.741464		3.093845		-4.5684361		
W11	2.852452		10.71693		2.75637388		
W12	24.94458		1.221686		-2.6894837		
W13	18.38504		11.91368		-3.5999347		
W14	1.817487		6.551981		-8.6374967		
W15	17.20405		-220.207		-6.867843		
W16	5.3589		-2.0084		-6.615674		
W17	15.93095		-4.40105		-6.3816806		
W18	8.642729		-1.97395		-1.8515056		
W19	8.531316		-8.86475		-1.7751005		
W20	7.147097		4.307462		10.5728455		
W21	-125.768		-5.92076		-8.8581685		
W22	0.978642		2.9824		-8.178778		
W23	0.236118		11.73002		-13.499183		
W24	12.22166		10.46385		-8.9498089		
W25	-0.01804		-0.22217		7.88692757		
W26	8.861057		3.506727		-3.225196		
W27	3.472298	W57	-15.927	W87	-4.8864959		
W28	8.995697	W58	-8.74706	W88	-7.3548948		
W29	9.605081	W59	0.762547	W89	-1.3747554		
W30	1.651695	W60	6.967352	W90	-2.8224927		

Lampiran 3 (Lanjutan)**Bobot dan bias Metode SVR dengan $C = 150$ dan $\sigma = 1.8$**

W1	1.952985	W31	17.40809	W61	-0.14193	W91	-7.65206
W2	8.996229	W32	20.0253	.	-9.82109	W92	-1.86716
W3	5.547252	.	3.027543	.	8.022645	W93	120.9391
W4	10.90721	.	3.652314	.	-8.50887		
W5	5.738499	.	13.86468	.	-5.18322		
W6	6.60789	.	8.754868	.	-10.1774		
W7	-10.2141	.	13.41708	.	-8.28953		
W8	3.584745	.	12.30406	.	-17.5802		
W9	-3.88876	.	4.932598	.	24.64755		
W10	6.859963	.	2.643498	.	-3.93291		
W11	2.518612	.	9.460288	.	2.223475		
W12	22.11045	.	1.041548	.	-2.27103		
W13	16.33913	.	10.48893	.	-3.12778		
W14	1.569989	.	5.797452	.	-7.62184		
W15	15.26117	.	-195.831	.	-6.03584		
W16	4.808175	.	-2.00297	.	-5.77806		
W17	14.15545	.	-4.03946	.	-5.59551		
W18	7.717037	.	-1.81124	.	-1.57417		
W19	7.596875	.	-8.07552	.	-1.49743		
W20	6.388444	.	3.722825	.	9.451872		
W21	-111.758	.	-5.33287	.	-7.79515		
W22	0.646166	.	2.475061	.	-7.27926		
W23	0.253112	.	10.34646	.	-11.8508		
W24	10.96266	.	9.185056	.	-7.84339		
W25	-0.03067	.	-0.34816	.	7.12679		
W26	7.825315	.	3.033229	.	-2.76281		
W27	3.117268	W57	-14.1478	W87	-4.35741		
W28	7.96312	W58	-7.82976	W88	-6.43258		
W29	8.543965	W59	0.732914	W89	-1.10906		
W30	1.443321	W60	6.185043	W90	-2.40003		

Lampiran 3 (Lanjutan)

Bobot dan bias Metode SVR dengan $C = 110$ dan $\sigma = 1.4$

W1	4.486584	W31	10.34052	W61	-6.20339	W91	-5.58958
W2	5.118372	W32	6.520148	.	-4.00237	W92	-1.2247
W3	-7.47834	.	9.856285	.	-7.4737	W93	90.96589
W4	2.761021	.	9.09178	.	-6.11938		
W5	-2.79899	.	3.506775	.	-13.082		
W6	5.129536	.	1.766094	.	18.52436		
W7	1.868701	.	6.949676	.	-2.70787		
W8	16.47809	.	0.682401	.	1.30924		
W9	12.25452	.	7.669313	.	-1.4806		
W10	1.096451	.	4.276446	.	-2.19149		
W11	11.37025	.	-147.051	.	-5.58959		
W12	3.699003	.	-2.00374	.	-4.38028		
W13	10.62333	.	-3.24398	.	-4.12959		
W14	5.856836	.	-1.48946	.	-4.03484		
W15	5.731432	.	-6.4763	.	-1.02911		
W16	4.86122	.	2.527444	.	-0.95511		
W17	-83.74	.	-4.18806	.	7.213078		
W18	0.006198	.	1.485045	.	-5.67521		
W19	0.271065	.	7.559691	.	-5.44086		
W20	8.397382	.	6.647673	.	-8.60684		
W21	-0.06123	.	-0.54519	.	-5.66021		
W22	5.763606	.	2.115239	.	5.574287		
W23	2.372938	.	-10.5978	.	-1.85732		
W24	5.904352	.	-5.97039	.	-3.25487		
W25	6.403545	.	0.636476	.	-4.61525		
W26	1.027201	.	4.623786	.	-0.60647		
W27	12.96927	W57	-0.21414	W87	-4.35741		
W28	15.06192	W58	-7.45079	W88	-6.43258		
W29	2.100309	W59	6.062642	W89	-1.10906		
W30	2.46989	W60	-6.20339	W90	-2.40003		

Lampiran 4
Bobot dan bias Metode PSO-SVR dengan $C = 102.2522$ dan
 $\sigma = 1.9724$

W1	18,280	W31	106,765	W61	-0.5526	W91	-41,485
W2	64,017	W32	126,225	.	-64,961	W92	-0.3299
W3	41,578	.	11,474	.	54,760	W93	783,207
W4	73,949	.	12,393	.	-49,724		
W5	40,465	.	83,792	.	-33,558		
W6	46,328	.	52,481	.	-58,692		
W7	-60,595	.	77,749	.	-47,252		
W8	28,275	.	71,413	.	-107,023		
W9	-18,281	.	23,249	.	162,168		
W10	48,021	.	0.8227	.	-21,788		
W11	17,891	.	54,140	.	0.6618		
W12	141,290	.	0.1914	.	-0.7405		
W13	103,865	.	59,820	.	-12,896		
W14	0.6283	.	32,448	.	-42,227		
W15	94,890	.	-126	.	-30,977		
W16	33,748	.	-31,128	.	-27,973		
W17	92,561	.	-31,120	.	-27,955		
W18	50,527	.	-15,120	.	-0.2635		
W19	49,783	.	-69,771	.	-0.2235		
W20	42,814	.	11,896	.	66,624		
W21	-710,860	.	-46,030	.	-43,676		
W22	-0.6036	.	0.0956	.	-42,137		
W23	0.1171	.	54,458	.	-64,504		
W24	72,470	.	45,443	.	-39,713		
W25	-0.3466	.	-15,160	.	55,642		
W26	45,153	.	13,156	.	-0.9841		
W27	18,473	W57	-91,437	W87	-24,207		
W28	47,363	W58	-50,690	W88	-32,177		
W29	52,215	W59	0.7709	W89	0.2770		
W30	0.5538	W60	38,287	W90	-0.6141		

Lampiran 5

Listing Program PSO-SVR

```

clc
clear all
n=input('Jumlah Partikel = ');
max_it=input('Jumlah Maksimum Iterasi = ');
a1=1; b1=2;
p1=a1+(b1-a1)*rand(n,1);
a2=100; b2=200;
C=a2+(b2-a2)*rand(n,1);
x=[C p1];
%Input Data Kurs Euro terhadap Dollar
dataset = xlsread('data1','sheet1');
for i=1:116
    for j=1:5
        datasetsusun(i,j)=dataset(i+j-1);
    end
end
%Normalisasi Data
dataskala=normalisasi(dataset);
% susuntabel data
for i=1:116
    for j=1:5
        susun(i,j)=dataskala(i+j-1);
    end
end
X=susun(1:93,1:4);
Y=susun(1:93,5);
ker='rbf';
loss='quadratic';
e=1;
for i=1:n
    %Peramalan menggunakan SVR
    [nsv, beta, bias] = svr(X,Y,ker,x(i,1),loss,e,x(i,2));
    nsv
    beta
    bias
    trnX = susun(1:93,1:4);
    tstX = susun(1:93,1:4);
    tstY(:,i) = svroutput(trnX,tstX,ker,beta,bias)
end
%Fungsi Evaluasi
[mape,px_best,pf_best,gx_best,gf_best]=evall(Y,tstY,x,n)
it=1;
while(it<=max_it)
    %Update kecepatan dan posisi
    [x_it,v_it]=update(px_best,gx_best,x,n);
    %evaluasi nilai fungsi tujuan
    for i=1:n

```

Lampiran 5(lanjutan)

```
[nsv, beta, bias] = svr(X,Y,ker,x_it(i,1),loss,e,x_it(i,2));
trnX = susun(1:93,1:4);
tstX = susun(1:93,1:4);
tstY(:,i) = svroutput(trnX,tstX,ker,beta,bias)
end
%Fungsi Evaluasi

[mape,px_bestit,pf_bestit,gx_bestit,gf_bestit]=evall(Y,tstY,
x_it,n);
for i=1:n
if px_bestit(i,3)<px_best(i,3)
    px_best(i,:)=px_bestit(i,:);
    x(i,:)=x_it(i,:);
else
    px_best(i,:)=px_best(i,:);
    x(i,:)=x(i,:);
end
end
%update gxbest dan gfbest
gx_best=sortrows(px_best,3);
gx_best=gx_best(1,:);
px_best_it(:, :,it)=px_best;
gf_best_it(:, :,it)=gf_bestit;
viter(:, :,it)= v_it;
xiter(:, :,it)= x_it;
iter(it, :)=gf_best_it(:, :,it);
it=it+1;
end
save('jumit.mat', 'iter');
[nsv, beta, bias] =
svr(X,Y,ker,gx_best(1,1),loss,e,gx_best(1,2));
x=[C p1]
beta
px_best_it
gf_best_it
gx_best
viter
xiter
trnX = susun(1:93,1:4);
tstA= susun(94:116,1:4);
tstY = svroutput(trnX,tstA,ker,beta,bias);
%denormalisasi
mindata=min(dataset(:));
maxdata=max(dataset(:));
rentang=maxdata-mindata;
denormalisasi= (tstY*rentang)+ mindata
%MAPE
aktual=datatasetsusun(94:116,5)
```

Lampiran 5(lanjutan)

```

a=(aktual-denormalisasi);
dx=abs(a/aktual)
fx=dx(1:23,5)
mape = (sum(fx)./23)*100
%plot
as=denormalisasi';
ad=aktual';
[brs,klm]=size(aktual)
plot(1:1:brs,ad(1,:), 'r-',1:1:brs,as(1,:), 'b-')
legend('aktual','peramalan')

```

Lampiran 6

Listing Program Training SVR

```

function [nsv, beta, bias] = svr(X,Y,ker,C,loss,e,pl)
    if (nargin < 3 | nargin > 7)
        help svr
    else
        fprintf('Support Vector Regressing ....\n')
        fprintf('_____ \n')
        n = size(X,1);
        if (nargin<6) e=0.0; end
        if (nargin<5) loss='eInsensitive'; end
        if (nargin<4) C=Inf; end
        if (nargin<3) ker='linear'; end
        % tolerance for Support Vector Detection
        epsilon = svtol(C);
        % pembuatan matriks kernel
        fprintf('Constructing ...\n');
        H = zeros(n,n);
        for i=1:n
            for j=1:n
                H(i,j) = svkernel(ker,X(i,:),X(j,:),pl);
            end
        end
        % Set up the parameters for the Optimisation problem
        switch lower(loss)
            case 'einsensitive',
                Hb = [H -H; -H H];
                c = [(e*ones(n,1) - Y); (e*ones(n,1) + Y)];
                vlb = zeros(2*n,1);
                vub = C*ones(2*n,1);
                x0 = zeros(2*n,1)
                neqcstr = nobias(ker);
                if neqcstr
                    A = [ones(1,n) -ones(1,n)];, b = 0
                    A = [];, b = [];
                else

```

Lampiran 6 (lanjutan)

Listing program training SVR

```

end
case 'quadratic',
    Hb = H + eye(n)/(2*C);
    c = -Y;
    vlb = -1e30*ones(n,1);
    vub = 1e30*ones(n,1);
    x0 = zeros(n,1);
    neqcstr = nobias(ker);
    if neqcstr
        A = ones(1,n);, b = 0;
    else
        A = [];, b = [];
    end
end

otherwise, disp('Error: Unknown Loss Function\n');
end

Hb = Hb+1e-10*eye(size(Hb));
% Penyelesaian optimasi
fprintf('Optimising ...\n');
st = cputime;
[alpha lambda how] = qp(Hb, c, A, b, vlb, vub, x0,
neqcstr);
fprintf('Execution time : %4.1f seconds\n',cputime -
st);
fprintf('Status : %s\n',how);
switch lower(loss)
    case 'einsensitive',
        beta = alpha(1:n) - alpha(n+1:2*n);
    case 'quadratic',
        beta = alpha;
end
fprintf('|w0|^2      : %f\n',beta'*H*beta);
fprintf('Sum beta : %f\n',sum(beta));
% menghitung support vector
svi = find( abs(beta) > epsilon );
nsv = length( svi );
fprintf('Support Vectors : %d
(%3.1f%%)\n',nsv,100*nsv/n);
bias = 0;
if nobias(ker) ~= 0
    switch lower(loss)
        case 'einsensitive',

```

Lampiran 6 (lanjutan)

Listing program training SVR

```

svii = find( abs(beta) > epsilon & abs(beta) < (C -
epsilon));
    if length(svii) > 0
        bias = (1/length(svii))*sum(Y(svii) -
e*sign(beta(svii)) - H(svii,svi)*beta(svi));
    else
        fprintf('No support vectors with interpolation
error e - cannot compute bias.\n');
        bias = (max(Y)+min(Y))/2;
    end
    case 'quadratic',
        bias = mean(Y - H*beta);
    end
end
end

end

```

Lampiran 7

Listing Program Testing SVR

```

function tstY = svroutput(trnX,tstX,ker,beta,bias)
    if (margin ~= 5)
        help svroutput
    else
        n = size(trnX,1);
        m = size(tstX,1);
        H = zeros(m,n);
        for i=1:m
            for j=1:n
                H(i,j) = svkernel(ker,tstX(i,:),trnX(j,:));
            end
        end
        tstY = (H*beta +bias);
    end
end

```

Lampiran 8

Listing program normalisasi

```

function dataskala=normalisasi(dataset);

mindata = min(dataset);
maxdata = max(dataset);
minskala = 0;
maxskala = 1;
% global normal;

```

```
dataskala = ((dataset-mindata)/(maxdata-mindata))*(maxskala-
minskala)+minskala;
```

Lampiran 9

Listing program evaluasi fitness

```
function
[mape,px_best,pf_best,gx_best,gf_best]=evall(Y,tstY,x,n);

for i=1:n
a=(Y-tstY(:,i));
bx=abs(a/Y);
dx=bx(1:93,4);
%error dibagi dengan target
mape(i,1)= (sum(dx)./93)*100;
end

pf_best=mape;
px_best=[x mape];
gx_best=sortrows(px_best,3);
gx_best=gx_best(1,:);
gf_best=gx_best(1,3);

return
```

Lampiran 10

Listing program update kecepatan partikel

```
function [x,v]=update(px_best,gx_best,x,n)
c1=2; c2=2;
r1=rand; r2=rand;
v(n,1)=[0];
v(n,2)=[0];
for i=1:n
v(i,:)=v(i)+(c1*r1).*(px_best(i,1:2)-
x(i,:))+(c2*r2).*(gx_best(1,1:2)-x(i,:));
end
x=x+v;
return
```

BIODATA PENULIS



Dilahirkan dengan nama Ade Rhinaldy, penulis sering dipanggil dengan nama pendek Ade. Penulis dilahirkan di Palangkaraya 4 Januari 1994, merupakan anak kedua dari tiga bersaudara. Penulis telah menempuh pendidikan formal yaitu SDN Palangka 6, SMPN 1 Palangkaraya, SMAN 1 Banjarmasin dan menyelesaikan program studi S1 jurusan Matematika ITS. Di jurusan matematika ITS penulis mengambil bidang minat Matematika Terapan. Penulis aktif di beberapa organisasi intra kampus yaitu Himpunan Mahasiswa Matematika (HIMATIKA) sebagai kepala divisi Olimpiade Matematika ITS 2014 (OMITS 2014). Dilingkup ITS Penulis juga terdaftar sebagai panitia aktif TEDxITS 2.0 dan TEDxITS 3.0.

Informasi mengenai Tugas Akhir ini dapat ditunjukkan ke alamat *email* rhinaldyade@gmail.com

“Halaman ini sengaja dikosongkan”

DAFTAR GAMBAR

Gambar 2.1 Bentuk Model SVR	13
Gambar 2.2 Pemetaan kernel mengubah permasalahan nonlinear menjadi linear	14
Gambar 3.1 Diagram alir metode SVR	25
Gambar 3.2 Diagram Alir Algoritma PSO-SVR	27
Gambar 4.1 Perbandingan data testing dan aktual	27
Gambar 4.2 Pergerakan nilai MAPE pada proses <i>training</i> PSO-SVR	43
Gambar 4.3 Plot data testing dan data aktual	45

“Halaman ini sengaja dikosongkan”

Peramalan Kurs EUR/USD Pada Pasar *Forex Online* Menggunakan *Support Vector Regression* Yang Dioptimasi Dengan Algoritma *Particle Swarm Optimization*

Ade Rhinaldy dan M. Isa Irawan

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Teknologi Sepuluh Nopember (ITS) Surabaya

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: rhinaldyade@gmail.com dan mii@its.ac.id

Abstrak— Pasar *forex online* merupakan pasar uang yang memiliki tingkat likuiditas yang tinggi. Hal ini menyebabkan harga kurs EUR/USD pada pasar *forex online* bergerak fluktuatif. Para pelaku pasar *forex* memanfaatkan hal ini untuk mendapatkan keuntungan dengan cara melakukan posisi jual pada saat harga naik dan posisi beli pada saat harga turun. Untuk memudahkan trader dalam mengambil keputusan jual dan beli dibutuhkan peramalan data harga kurs EUR/USD. Penelitian ini bertujuan untuk memprediksi arah pergerakan harga harian perdagangan *forex* dengan menggunakan *support vector regression* (SVR). Parameter pada SVR di optimasi dengan menggunakan algoritma *particle swarm optimization* (PSO) sehingga didapatkan hasil peramalan yang baik. Hasil dari penelitian ini didapatkan hasil peramalan dengan nilai MAPE = 0.1895% yang dihasilkan dengan memasukkan nilai parameter optimal *penalty* dari SVR (C) = 102.2522 dan parameter kernel RBF (σ) = 1.9724.

Kata Kunci— Peramalan, *forex*, *Support Vector Regression*, *Particle Swarm Optimization*.

I. PENDAHULUAN

Perdagangan valas atau *forex* merupakan kegiatan perdagangan mata uang dari negara yang berbeda satu sama lain. Pada pasar *forex* yang diperdagangkan adalah mata uang dengan pelakunya yang sangat luas meliputi bank, perusahaan besar, negara, institusi, spekulasi bahkan perorangan. Biasanya pelaku perdagangan pasar *forex* yang mencakup seluruh negara yang ada di dunia menjadikan pasar *forex* sebagai pasar uang yang paling besar dan sangat likuid. Pelaku pasar *forex* yang ada di seluruh dunia ini juga menyebabkan pasar *forex* buka 24 jam selama lima hari dalam seminggu, sehingga pelakunya dapat melakukan transaksi kapanpun.

Pasar *forex* merupakan pasar yang memiliki kelebihan di tingkat likuiditas yang tinggi pada waktu ke waktu. Tingginya likuiditas menyebabkan harga di pasar mata uang bergerak sangat fluktuatif. Hal ini menyebabkan *forex trading* menjadi investasi pada sektor finansial yang tergolong paling *high risk-high return investment* [1].

Tingginya fluktuasi harga *forex* menyebabkan para pedagang *forex* (*trader*) harus mengetahui waktu yang tepat untuk mengambil keputusan kapan melakukan penjualan dan pembelian. Untuk membantu para *trader* dalam menentukan perencanaan, maka kajian matematika yaitu peramalan (*forecasting*) dapat digunakan untuk membantu permasalahan ini.

Pada permasalahan peramalan (*forecasting*) data, terdapat berbagai macam metode untuk mendapatkan hasil ramalan. Salah satu metode yang dapat digunakan adalah metode *support vector regression* (SVR) yang merupakan cabang dari *support vector machine* (SVM). SVM muncul sebagai suatu teknik baru dan kuat dalam proses pembelajaran mesin (*machine learning*) dari data dan secara khusus dalam penyelesaian permasalahan klasifikasi dan SVR sebagai penyelesaian permasalahan regresi [2]. SVR merupakan metode pembelajaran mesin yang sangat cocok untuk penyelesaian

permasalahan dengan data yang memiliki tingkat fluktuasi data tinggi [2]. Untuk mendapatkan model peramalan yang optimal dari metode SVM, maka penting untuk memilih fungsi kernel, mengatur parameter kernel dan menentukan margin nilai penalti (C) [3]. Xibin Wang dkk. dalam tulisan ilmiahnya [4] mengatakan teknik optimasi yang paling umum adalah algoritma koloni semut, algoritma *gridsearch* dan algoritma genetika. algoritma koloni semut memiliki kelemahan diantaranya langkanya feromon awal, waktu pencarian yang lama dan memiliki solusi terbaiknya bersifat lokal; algoritma *gridsearch* adalah algoritma yang bersifat komputasi intensif, memakan waktu yang lama dan akurasi pembelajar (*training*) rendah; algoritma genetika adalah algoritma yang operasinya kompleks dan dibutuhkan isu-isu yang berbeda untuk merancang *crossover* yang berbeda atau mutasi. *Particle swarm optimization* (PSO) merupakan sebuah algoritma dengan konsep sederhana tetapi sangat efektif dalam mengoptimasi permasalahan berbagai fungsi, selain itu algoritma PSO mudah dalam implementasi dan memiliki konvergensi yang cepat [5].

Berdasarkan latar belakang diatas, maka penulis memutuskan untuk mengangkat judul Peramalan Kurs EUR/USD Pada Pasar *Forex Online* Menggunakan *Support Vector Regression* Yang Dioptimasi Dengan Algoritma *Particle Swarm Optimization*.

II. KAJIAN TEORI

A. Peramalan Kurs EUR/USD Pada Pasar *Forex Online*

Pasar valuta asing (bahasa Inggris: *foreign exchange market, forex*) atau disingkat valas merupakan suatu jenis perdagangan atau transaksi yang memperdagangkan mata uang suatu negara terhadap mata uang negara lainnya (pasangan mata uang/pair) yang melibatkan pasar-pasar uang utama di dunia selama 24 jam sehari dan lima hari dalam seminggu secara berkesinambungan.

Peramalan adalah proses untuk memperkirakan berapa kebutuhan dimasa mendatang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan untuk memenuhi permintaan barang atau jasa [6].

Peramalan kurs EUR/USD pada pasar *forex online* digunakan untuk merkirakan harga kurs EUR/USD yang bergerak fluktuatif. Sehingga para trader dapat mengambil keputusan kapan untuk membeli dana kapan untuk menjual.

B. *Support Vector Regression* (SVR)

Support vector regression (SVR) merupakan cabang dari metode pembelajaran mesin *support vector machine* (SVM) dalam permasalahan regresi dan peramalan *time series* dikenal sebagai SVR[7].

Misalkan diberikan set data pembelajaran, (x_i, y_i) dimana $i = 1, \dots, n$ dengan data *input* $x = \{x_1, \dots, x_n\} \in \mathcal{R}^N$ dan data *output* $y = \{y_1, \dots, y_n\} \in \mathcal{R}$. Dengan SVR, kita hendak menemukan suatu fungsi $f(x)$ yang mempunyai deviasi (ϵ) paling besar dari target aktual y_i untuk semua data *training*. Sehingga dengan SVR akan didapatkan suatu tabung seperti Gambar 2.1.

Model regresi mendefinisikan hubungan antara x_i dan $f(x_i)$ sebagai berikut [7] :

$$y = f(x_i) = (wx_i) + b \quad (1)$$

dimana w adalah parameter bobot, b adalah parameter bias. Koefisien w dan b akan diestimasi dengan persamaan optimasi linear berikut :

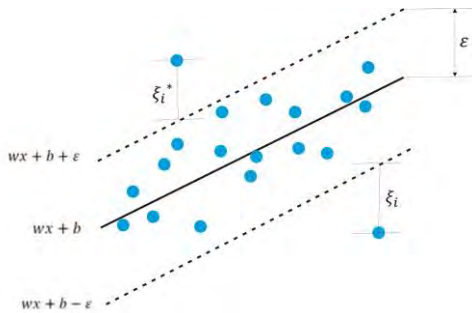
$$\min_{w,b,\xi^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2)$$

dengan kendala

$$\begin{aligned} ((w \cdot x_i) + b) - y_i &\leq \varepsilon + \xi_i, & i = 1, \dots, n \\ y_i - ((w \cdot x_i) + b) &\leq \varepsilon + \xi_i^*, & i = 1, \dots, n \\ \xi_i^* &\geq 0 \end{aligned} \quad (3)$$

Pada persamaan diatas, ξ merupakan *variable slack*, C adalah koefisien pinalti dan ε adalah *insensitive loss function* atau dalam SVR, ε ekuivalen dengan akurasi dari aproksimasi kita terhadap data *training*. C mengontrol keseimbangan antara kompleksitas model dan error dari *training*; ε adalah konstanta yang telah ditetapkan bahwa untuk mengendalikan regresi.

Permasalahan optimasi diatas dimana kendala adalah permasalahan *quadratic programming* dapat diselesaikan dengan mentransformasikan permasalahan tersebut ke bentuk dual.



Gambar 2.1 Bentuk model SVR

Dengan menggunakan pengali lagrange α_i dimana α_i merupakan *variable non-negatif*, maka bentuk primal pemrograman kuadratik (2) dapat diubah menjadi bentuk dual sebagai berikut :

$$\begin{aligned} \max \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i') (\alpha_i - \alpha_i') K(x_i x_j) + \\ \sum_{i=1}^n y_i ((\alpha_i - \alpha_i') - \sum_{i=1}^n (\alpha_i + \alpha_i')) \geq 0 \end{aligned} \quad (4)$$

dengan kendala

$$\begin{aligned} \sum_{i=1}^n (\alpha_i - \alpha_i') &= 0 \\ 0 \leq \alpha_i \leq C, i &= 1, \dots, n \end{aligned} \quad (5)$$

Pada persamaan diatas, C didefinisikan oleh user.

Untuk mengatasi permasalahan nonlinear maka kita gunakan metode kernel. $K(x_i, x_j)$ adalah dot-product kernel yang didefinisikan sebagai $K(x_i, x_j) = \Phi^T(x_i) \Phi(x_j)$. Sehingga fungsi regresi dirumuskan sebagai berikut :

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i') K(x_i, x_j) + \quad (6)$$

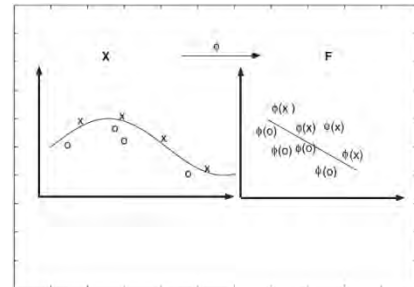
Metode kernel [8] memberikan pendekatan alternatif dengan cara melakukan *mapping* data x dari input space ke *feature space* F melalui suatu fungsi ϕ sehingga $\phi: x \rightarrow \phi(x)$. Karena itu suatu titik x pada input *space* menjadi $\phi(x)$ di *feature space*. Pemetaan input *space* ke *feature space* dapat dilihat pada Gambar 2.2. Hal ini menyebabkan $K(x, x') = \phi(x) \cdot \phi(x')$ dimana (\cdot) merupakan operasi perkalian dalam. Kemudian kita bisa membuat suatu fungsi pemisah yang linier yang mewakili fungsi nonlinear di *input space*.

Penggunaan fungsi kernel yang berbeda akan menghasilkan model regresi yang berbeda pula. Fungsi kernel yang biasanya dipakai dalam literatur SVM [8]:

$$\begin{aligned} \text{Kernel linear} &: K(x, x') = x^T x \\ \text{Polynomial} &: K(x, x') = (x^T x + 1)^p \end{aligned}$$

$$\text{Radial Basis Function (RBF)} : K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Pada permasalahan peramalan kali ini, penulis menggunakan fungsi kernel RBF, dimana σ adalah parameter dari kernel. Parameter σ mendefinisikan struktur tingkat dimensi *feature space*. Pada penjabaran di atas kita ketahui bahwa pemilihan dari parameter C , dan σ memiliki pengaruh yang besar terhadap keakuratan peramalan. Oleh karena itu algoritma PSO digunakan untuk mendapatkan parameter dari SVR yang optimal.



Gambar 2.2 Pemetaan kernel mengubah permasalahan nonlinear menjadi linear

C. Particle Swarm Optimization (PSO)

Particle Swarm Optimization merupakan metode optimasi yang pertama kali dikembangkan oleh James Kennedy dan Russell Eberhart [5]. PSO merupakan metode optimasi dimana algoritmanya mengadopsi pada perilaku dan psikologi sosial sebuah organisme seperti rayap, lebah atau burung [9]. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya.

Algoritma dari PSO ditunjukkan sebagai berikut :

1. Inisialisasi parameter PSO, yaitu menentukan populasi kandidat solusi dari partikel untuk parameter σ (parameter kernel RBF) dan C (nilai pinalti)
2. Menentukan vektor posisi $X_j(i) = X_1(i), X_2(i), \dots, X_N(i)$ dan vektor kecepatan $V_j(i) = V_1(i), V_2(i), \dots, V_N(i)$, dimana $i = 0, 1, 2, \dots, \text{batas iterasi}, j = 1, 2, \dots, N$ dan $N =$ banyaknya partikel.
3. Mendapatkan nilai kelayakan (*current fitness*) dari tiap-tiap kandidat solusi atau partikel dengan mengevaluasi nilai fungsi tujuan dimana pada penelitian ini fungsi tujuannya adalah meminimalkan nilai MAPE peramalan. Evaluasi nilai fungsi tujuan untuk setiap partikel pada $i = 0$ dan nyatakan dengan

$$f|X_1(0)|, f|X_2(0)|, \dots, f|X_N(0)|$$

4. Dari perhitungan fungsi objektif, dapat diketahui *local best fitness* (Pf_best) dan *local best position* (Px_best).
5. Mencari nilai *global best fitness* (Gf_best), yaitu nilai yang minimum dari *local best fitness* (Pf_best).
6. Menentukan *global best position* (Gx_best), yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* (Px_best) dari partikel yang memenuhi persyaratan *global best fitness* (Gf_best).
7. Memperbarui kecepatan dengan menggunakan persamaan

$$V_j(i) = V_j(i-1) + c_1 r_1 [P_{best,j} - x_j(i-1)] + c_2 r_2 [G_{best} - x_j(i-1)],$$

$$j = 1, 2, \dots, N \quad (7)$$

dimana c_1 dan c_2 masing-masing adalah *learning rates* untuk kemampuan individu (*cognitive*) dan pengaruh sosial (*group*), r_1 dan r_2 bilangan random yang berdistribusi uniform dalam interval 0 dan 1. Memperbarui posisi partikel selanjutnya dihitung dengan cara

$$X_j(i) = X_j(i-1) + V_j(i); j = 1, 2, \dots, N \quad (8)$$

8. Ulangi langkah 2 sampai 7 hingga fungsi objektif konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi

III. ANALISIS DATA DAN PEMBAHASAN

Gambaran sistem peramalan data kurs *closing* EUR/USD pada pasa *forex online* dengan menggunakan metode PSO-SVR memiliki beberapa tahapan yakni, pengambilan data, prapemrosesan data peramalan, *training* data, dan *testing* data.

A. Pengambilan Data

Data yang digunakan dalam penulisan tugas akhir ini adalah data dari kurs *closing* EUR/USD pasar *forex online*. Data diambil setiap satu jam dengan bantuan software trading Metatrader 4. Data historis diambil pada tanggal 12 Januari 2014 sampai 16 Januari 2014 sehingga data sebanyak 120 data yang diambil. Data kemudian diolah dengan *software* Matlab, menggunakan metode *Particle Swarm Optimization – Support Vector Regression*.

B. Prapemrosesan Data

Berikut ini tahapan-tahapan untuk tahapan prapemrosesan data harga kurs EUR/USD :

1. Masukkan data kedalam Matlab, bagi data menjadi data *training*, dan data *testing*, dimana proporsi untuk data *training* yaitu 80% = 93 data dimulai dari data ke-1 hingga data ke-93 dan data *testing* 20% = 27 data dimulai dari data ke-94 hingga data ke-120.
2. SVR bekerja lebih efektif dengan data yang mempunyai rentang skala kecil. Oleh karena itu data peramalan harga *closing* kurs EUR/USD di normalisasi dengan rentang nilai antara 0 sampai 1. Dalam tugas akhir ini, metode yang dipilih untuk menormalisasi data adalah metode min-max. Persamaan metode min-max ini dapat dilihat pada persamaan (9) :

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \quad (9)$$

dengan :

- v'_i : data baru hasil normalisasi
- v_i : data input yang akan di normalisasi
- \min_A : nilai minimum dari data input
- \max_A : nilai maksimum dari data input
- new_min_A : nilai minimum dari data hasil

normalisasi yang diinginkan
 new_max_A : nilai maksimum dari data hasil
 normalisasi yang diinginkan

3. Setelah data dinormalisasi tahap selanjutnya adalah menyusun data *training* ke dalam beberapa fitur. Dalam penelitian ini penulis menggunakan 4 fitur. Pada tabel 1. ditunjukkan cara penyusunan data sebelum di proses.

Tabel 1. Bentuk penyusunan data

	f_1	f_2	f_3	f_4	<i>target</i>
<i>Training</i>	data_1	data_2	data_3	data_4	data_5
	data_2	data_3	data_4	data_5	data_6

	data_{k-4}	data_{k-3}	data_{k-2}	data_{k-1}	data_k
<i>Testing</i>	data_{k-3}	data_{k-2}	data_{k-1}	data_k	data_{k+1}
	data_{k-2}	data_{k-1}	data_k	data_{k+1}	data_{k+2}

	data_{n-4}	data_{n-3}	data_{n-2}	data_{n-1}	data_n

C. Peramalan dengan Menggunakan Metode SVR

a. Training

Training data merupakan proses untuk mendapatkan nilai bobot (w) dan bias (b). Nilai dari w dan b akan dimasukkan ke model regresi pada persamaan (1).

Peramalan dengan menggunakan metode SVR membutuhkan pengguna untuk menginputkan parameter yang dibutuhkan oleh SVR. Dalam penelitian ini peramalan dengan SVR menggunakan parameter $C = 110$ dan $\sigma = 1.8$ yang diambil secara *trial and error*. Langkah pertama dalam metode SVR adalah membuat matriks kernel dengan menggunakan fungsi kernel *Radial Basis Function* (RBF), dengan parameter RBF yang telah di inialisasi pada tahap sebelumnya. Berikut ini contoh dalam membuat matriks kernel.

$$\text{Kernel RBF} := \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Berikut ini contoh perhitungan fungsi kernel RBF pada indeks $i = 1, j = 1$ dengan menggunakan salah satu nilai partikel dari salah satu nilai pada parameter σ (RBF).

$$K(x_1, x_1) = \exp\left(-\frac{\| [0.9959 \ 1 \ 0.9972 \ 0.9698] - [0.9959 \ 1 \ 0.9972 \ 0.9698] \|^2}{2(1.5)^2}\right)$$

Setelah didapat nilai $K(x_1, x_1)$, selanjutnya nilai tersebut dibawa ke matriks dengan indeks $i = 1$ dan $j = 1$. Nilai indeks i dan j terus berjalan hingga matriks kernel terisi nilai dengan ukuran matriks yang dibentuk sebesar 93×93 .

$$\begin{bmatrix} K(x_1, x_1) & \dots & \dots & K(x_1, x_{93}) \\ \vdots & & & \\ K(x_{93}, x_1) & \dots & \dots & K(x_{93}, x_{93}) \end{bmatrix}$$

Menghitung nilai bobot dan bias dengan memasukkan nilai mapping dari fungsi kernel dan nilai C (koefisien pinalti) ke dalam *quadratic programming* (QP).

$$\max \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i') (\alpha_i - \alpha_i') K(x_i, x_j) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i') - \sum_{i=1}^n (\alpha_i + \alpha_i') \xi_i^* \geq 0$$

dengan kendala

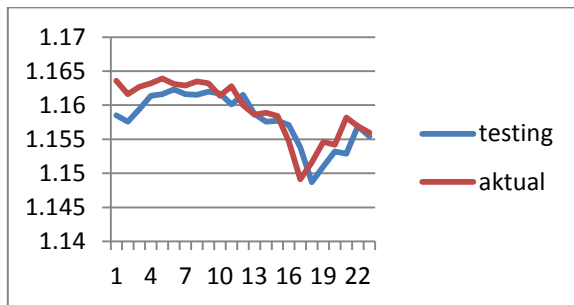
$$\sum_{i=1}^n (\alpha_i - \alpha_i') = 0 \\ 0 \leq \alpha_i \leq C, i = 1, \dots, n$$

b. Testing

Dari perhitungan *quadratic programming* didapat nilai bobot (w) dan bias (b). Dari nilai w dan b dihasilkan persamaan fungsi regresi sebagai berikut.

$$f(x) = (2.1979) \left(\exp \left(- \frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (7.1699) \left(\exp \left(- \frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots + (84.6638) \left(\exp \left(- \frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + 0$$

Untuk menguji apakah model yang terbentuk baik, maka dilakukan proses *testing*. Pada proses *testing* model yang terbentuk diuji dengan menggunakan data *testing* dan dihasilkan nilai peramalan dengan akurasi peramalan sebesar MAPE sebesar 0.2135%.



Gambar 3.1 Perbandingan data aktual dan data *testing* SVR

Pada Gambar 3.1 terlihat bahwa hasil peramalan dengan menggunakan metode SVR menghasilkan bentuk grafik yang serupa dengan pergerakan data aktual. Hal ini menunjukkan metode SVR dapat meramal data kurs EUR/USD pada pasar *forex online* dengan baik.

D. Training dan Testing SVR yang Dioptimasi Algoritma PSO

Nilai MAPE akan lebih baik jika nilai dari parameter C dan σ dioptimasi. Berikut ini tahapan perhitungan peramalan dengan menggunakan metode SVR yang dioptimasi dengan algoritma PSO.

a. Training

Tahapan *training* dalam penelitian ini adalah sebagai berikut :

- Menentukan ukuran atau jumlah partikel (N) dari parameter PSO yaitu parameter C (*penalty cost*) dan σ (RBF). Untuk mengurangi jumlah evaluasi fungsi yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar, tetapi juga tidak terlalu kecil, agar ada banyak kemungkinan posisi menuju solusi terbaik atau optimal. Jika terlalu kecil, sedikit kemungkinan menemukan posisi partikel yang baik.

- Terlalu besar juga akan membuat perhitungan jadi panjang. Biasanya digunakan ukuran kawanan adalah 20 sampai 30 partikel. Pada penelitian ini jumlah partikel $N = 20$, kemudian kita merandom nilai awal untuk setiap partikel dan didapat nilai awal partikel dari parameter C dan σ pada Tabel 2.

Tabel 2. Nilai awal partikel

Partikel	C	σ
1	167.8735	1.6324
2	175.7740	1.0975
3	174.3132	1.2785
4	139.2227	1.5469
5	165.5478	1.9575
6	117.1187	1.9649
7	170.6046	1.1576
8	103.1833	1.9706
9	127.6923	1.9572
10	104.6171	1.4854
11	109.7132	1.8003
12	182.3458	1.1419
13	169.4829	1.4218
14	131.7099	1.9157
15	195.0222	1.7922
16	103.4446	1.9595
17	143.8744	1.6557
18	138.1558	1.0357
19	176.5517	1.8491
20	179.5200	1.9340

- Dari partikel yang didapat pada tahap sebelumnya, selanjutnya kita hitung nilai peramalannya dengan metode SVR dan kemudian dihitung nilai akurasi. Dari perhitungan SVR didapat nilai bobot (w) dan bias (b). Setelah didapat nilai bobot (w) dan bias (b) kita *testing* nilai tersebut dengan menggunakan data *training* sebagai data *testing*. Dari proses ini didapatkan nilai ramalan kemudian kita hitung nilai MAPEnya dan dijadikan sebagai fungsi tujuan atau fungsi objektif dari algoritma PSO. Hitung fungsi objektif yaitu :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - F_i}{x_i} \right| \times 100\% \quad (10)$$

Perhitungan fungsi objektif digunakan untuk mendapatkan nilai kelayakan (*current fitness*) dari tiap-tiap kandidat solusi atau partikel.

- Perhitungan fungsi objektif, dapat diketahui *local best fitness* (Pf_best) dan *local best position* (Px_best) yang dapat dilihat pada Tabel 3.

Tabel 3. Nilai Px_best dan Pf_best

Partikel	Px_best C	Px_best σ	Pf_best C dan σ
1	167.8735	1.6324	3.5084
2	175.7740	1.0975	3.5124
3	174.3132	1.2785	3.5117
4	139.2227	1.5469	3.4909
5	165.5478	1.9575	3.5071
6	117.1187	1.9649	3.4777
7	170.6046	1.1576	3.5098
8	103.1833	1.9706	3.4675
9	127.6923	1.9572	3.4844
10	104.6171	1.4854	3.4686
11	109.7132	1.8003	3.4725

Partikel	Px_best C	Px_best σ	Pf_best C dan σ
12	182.3458	1.1419	3.5156
13	169.4829	1.4218	3.5092
14	131.7099	1.9157	3.4868
15	195.0222	1.7922	3.5212
16	103.4446	1.9595	3.4677
17	143.8744	1.6557	3.4939
18	138.1558	1.0357	3.4903
19	176.5517	1.8491	1.8491
20	179.5200	1.9340	3.5143

j	$X_j(1) C$	$X_j(1) \sigma$
11	103.3170	1.9671
12	104.8038	1.9536
13	104.5405	1.9594
14	103.7673	1.9695
15	105.0633	1.9669
16	103.1886	1.9704
17	104.0163	1.9641
18	103.8992	1.9515
19	104.6852	1.9681
20	104.7460	1.9698

- Mencari nilai *global best fitness* (Gf_best), yaitu nilai yang minimum dari *local best fitness* (Pf_best). Dari perhitungan didapat nilai *global best fitness* (Gf_best) sebesar 3,6228.
- Menentukan *global best position* (Gx_best), yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* (Px_best). dari partikel yang memenuhi persyaratan *global best fitness* (Gf_best). Sehingga didapatkan partikel yang menghasilkan nilai *global best fitness* (Gf_best) sebesar 3,6228 adalah pada saat $C = 104.9654$ dan $\sigma = 1.9106$
- Memperbarui kecepatan dengan menggunakan persamaan (7). Kecepatan partikel pada iterasi 1 dapat dilihat pada Tabel 5. Setelah didapat kecepatan baru untuk setiap partikel maka langkah selanjutnya adalah memperbarui posisi partikel dengan persamaan (8). Sehingga didapat posisi terbaru pada Tabel 5.
- Setelah didapat posisi partikel yang baru, hitung kembali fungsi tujuan dengan menggunakan posisi partikel sekarang. Jika nilai pf_best sekarang lebih baik dari nilai pf_best dari iterasi sebelumnya, maka nilai px_best sebelumnya digantikan dengan px_best sekarang.
- Ulangi langkah 3 sampai 7 hingga fungsi objektif atau nilai Gf_best konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi.

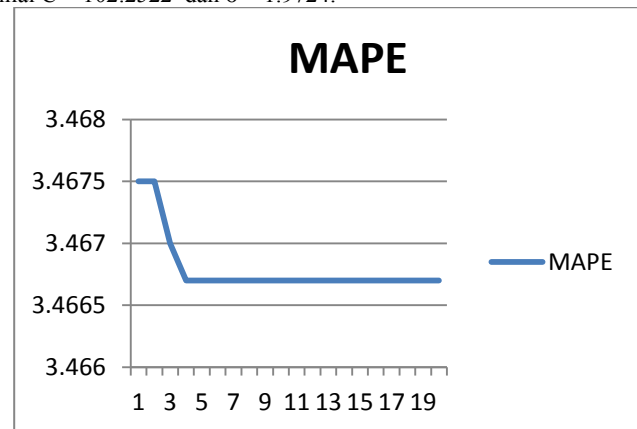
Pada penelitian ini nilai dari fungsi objektif telah konvergen pada iterasi ke-6. Maka didapatkan nilai masukkan C dan σ terbaik untuk mendapatkan nilai bobot (w) dan bias (b) yang paling baik pula. Pada Gambar 3.1 nilai MAPE pada proses *training* bergerak menuju nilai MAPE yang minimum disetiap iterasi. Nilai MAPE mulai konvergen pada iterasi ke-6 dan pada iterasi ke-15 didapatkan nilai $C = 102.2522$ dan $\sigma = 1.9724$.

Tabel 4. Kecepatan partikel pada iterasi 1

j	$V_j(1)C$	$V_j(1) \sigma$
1	-63.3659	0.3313
2	-71.1047	0.8552
3	-69.6738	0.6779
4	-35.3016	0.4150
5	-61.0878	0.0128
6	-13.6501	0.0056
7	-66.0411	0.7963
8	0	0
9	-24.0073	0.0132
10	-1.4045	0.4753
11	-6.3962	0.1668
12	-77.5419	0.8117
13	-64.9423	0.5376
14	-27.9427	0.0537
15	-89.9589	0.1747
16	-0.2560	0.0109
17	-39.8582	0.3084
18	-34.2566	0.9157
19	-71.8665	0.1190
20	-74.7740	0.0359

Tabel 5. Posisi partikel yang diperbarui

j	$X_j(1) C$	$X_j(1) \sigma$
1	104.5076	1.9637
2	104.6693	1.9527
3	104.6394	1.9564
4	103.9211	1.9619
5	104.4600	1.9703
6	103.4686	1.9705
7	104.5635	1.9540
8	103.1833	1.9706
9	103.6850	1.9703
10	103.2126	1.9607

Gambar 3.2 Pergerakan nilai MAPE *training*

Training data akan menghasilkan nilai bobot (w) dan bias (b) yang didapatkan dengan memasukkan parameter $C = 102.2522$ dan $\sigma = 1.9724$.

b. Testing

Untuk menguji nilai bobot (w) dan bias (b) maka dilakukan *testing* dengan data yang berbeda. Data yang digunakan untuk proses *testing* adalah data kurs EUR/USD pada periode 94-116. Dari proses ini kita lihat seberapa besar nilai eror yang dihasilkan dengan menggunakan bobot (w) dan bias (b) pada proses *training*.

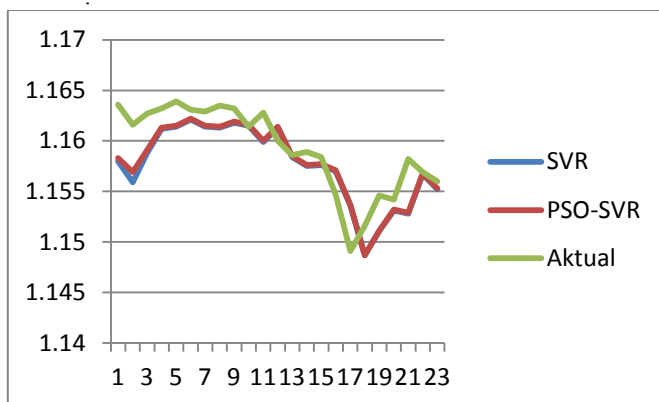
Nilai dari bobot (w) dan bias (b) pada proses *training* dimasukkan ke dalam fungsi regresi sehingga didapat model regresi sebagai berikut:

$$f(x) = \left((18.280) \left(\exp \left(-\frac{\|x_i - x_j^T\|^2}{2\sigma^2} \right) \right) + (64.017) \left(\exp \left(-\frac{\|x_i - x_j^T\|^2}{2\sigma^2} \right) \right) + \dots + (783.207) \left(\exp \left(-\frac{\|x_i - x_j^T\|^2}{2\sigma^2} \right) \right) \right) + 0$$

Hasil dari proses testing berupa nilai ramalan dapat dilihat pada Tabel 6. Keakuratan hasil ramalan pada proses testing dihitung dengan menghitung nilai MAPE menggunakan persamaan (10). Perhitungan tersebut menghasilkan nilai MAPE sebesar 0.1895%. Pada Gambar 3.2 berikut ini ditunjukkan perbandingan hasil peramalan pada proses testing terhadap data aktual kurs closing EUR/USD pada pasar *forex online*.

Tabel 6. Perbandingan data aktual dan data peramalan

Periode	Aktual	Peramalan
94	1.1636	1.1583
95	1.1616	1.1569
96	1.1627	1.1591
97	1.1632	1.1613
98	1.1639	1.1615
99	1.1631	1.1622
100	1.1629	1.1615
101	1.1635	1.1614
102	1.1632	1.1619
103	1.1614	1.1616
104	1.1628	1.1600
105	1.1600	1.1614
106	1.1586	1.1585
107	1.1589	1.1576
108	1.1584	1.1577
109	1.1547	1.1571
110	1.1491	1.1537
111	1.1516	1.1487
112	1.1546	1.1511
113	1.1542	1.1532
114	1.1582	1.1529
115	1.1569	1.1568
116	1.1560	1.1553



Gambar 3.3 Perbandingan Metode SVR dan PSO-SVR

Dari Gambar 3.3 terlihat bahwa bentuk pergerakan grafik data peramalan PSO-SVR menyerupai pergerakan grafik data aktual. Metode PSO-SVR menghasilkan nilai MAPE yang sangat kecil dan

lebih kecil dibandingkan peramalan dengan menggunakan metode SVR. Maka dapat ditarik kesimpulan model PSO-SVR yang didapat pada proses *training* memiliki keakuratan yang baik daripada model yang dihasilkan metode SVR saat di uji dengan menggunakan data *testing*. Hal ini juga menunjukkan bahwa algoritma PSO berhasil untuk mengoptimasi parameter dari SVR.

IV. KESIMPULAN

Berdasarkan analisis terhadap hasil perhitungan peramalan yang telah dilakukan terhadap data closing kurs EUR/USD pada pasar *forex online* dengan menggunakan metode *Particle Swarm Optimization - Support Vector Regression*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Parameter dari *support vector regression* yang telah dioptimasi dengan menggunakan *particle swarm optimization* optimal pada saat C bernilai 102.2522 dan σ bernilai 1.9724.
2. Tugas Akhir ini telah berhasil melakukan peramalan data kurs EUR/USD pada pasar *forex online* menggunakan metode peramalan support vector regression dengan nilai parameter C = 110 dan $\sigma = 1.5$ didapatkan MAPE sebesar 0.2135%. Untuk hasil peramalan menggunakan support vector regression dengan parameter yang telah optimal C = 102.2522 dan $\sigma = 1.9724$ didapatkan MAPE sebesar 0.1895%.

DAFTAR PUSTAKA

- [1] Anonim. 2009. "Struktur Investasi". [http:// elajarforex.com/sitting-duck/1-struktur-investasi.html](http://elajarforex.com/sitting-duck/1-struktur-investasi.html). Diakses pada tanggal 6 Februari 2015.
- [2] Kamruzzaman, J. dan Ruhul A Sarker. 2004. "Aplication of Support Vector Machine to *Forex* Monitoring". IEEJ.
- [3] X. Liang, H. Zhang, J. Xiao, Y. Chen. 2009. "Improving option price forecasts with neuralnetworks and support vector regressions". *Neurocomputing* 72 (13–15) 3055–3065.
- [4] Wang, X, Junhao Wen, Yihao Zhang, Yu iao Wang. 2013. "Real Estate Price Forecasting Based on SVM Optimized y PSO". *Optik*, 125 (2014) 1439-1443.
- [5] J. Kennedy and R.C. E erhart.1995. "A New Optimizer Using Particle Swarm Theory".
- [6] Nasution, A. H. 1999. "Metode Aplikasi Peramalan (2 ed, Vol. 1)". Binarupa Aksara
- [7] Santosa, B. 2007. "Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta: Graha Ilmu.
- [8] B. Scholkopf and A.J. Smola. 2002. "Learning with Kernels". The MIT Press, Cambridge, Massachusetts.
- [9] Santosa, B. "Tutorial Particle Swarm Optimization". Teknik Industri ITS. Surabaya.



PERAMALAN KURS EUR/USD PADA PASAR *FOREX*
ONLINE MENGGUNAKAN SUPPORT VECTOR
REGRESSION YANG DIOPTIMASI DENGAN
ALGORITMA PARTICLE SWARM OPTIMIZATION

Ade Rhinaldy

1211100106

Calon Dosen Pembimbing

Prof. Dr. M. Isa Irawan, MT

MATEMATIKA ITS

2015

Abstrak

Pasar *forex online* merupakan pasar uang yang memiliki tingkat likuiditas yang tinggi. Hal ini menyebabkan harga kurs EUR/USD pada pasar *forex online* bergerak fluktuatif. Para pelaku pasar *forex* memanfaatkan hal ini untuk mendapatkan keuntungan dengan cara melakukan posisi jual pada saat harga naik dan membeli pada saat harga turun. Untuk memudahkan trader dalam mengambil keputusan jual dan beli dibutuhkan bantuan kajian matematika. Peramalan merupakan salah satu cabang ilmu matematika yang dapat digunakan sebagai alat bantu. Penelitian ini bertujuan untuk memprediksi arah pergerakan harga harian perdagangan forex dengan menggunakan support vector regression (SVR). Parameter pada SVR di optimasi dengan menggunakan algoritma Particle Swarm Optimization (PSO) sehingga didapatkan hasil peramalan yang baik. Hasil dari penelitian ini didapatkan hasil peramalan dengan tingkat eror sebesar 99,8105% yang dihasilkan dengan memasukkan nilai parameter optimal $C = 102.2522$ dan $\sigma = 1.9724$.

BAB I

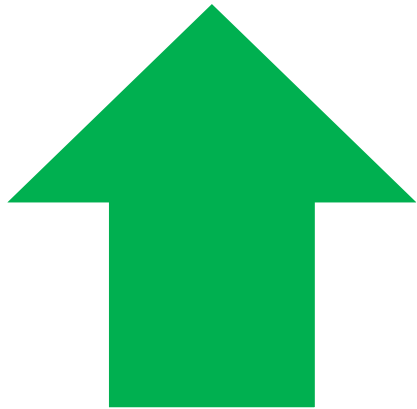
Pendahuluan



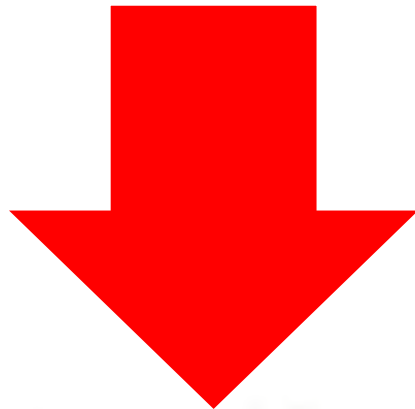
Pasar foreign exchange (forex) online merupakan pasar dimana yang dijual adalah mata uang suatu negara terhadap mata uang negara lainnya



Pasar forex online merupakan pasar uang yang memiliki tingkat likuiditas yang tinggi, sehingga harga di pasar mata uang bergerak fluktuatif



Jual/Sell



Beli/Buy



LATAR BELAKANG

Fluktuatif harga, menyebabkan pasar forex menjadi investasi yang bersifat *high risk-high return investment*

Kajian matematika yang dapat digunakan dalam permasalahan ini adalah peramalan

Metode peramalan yang digunakan
*particle swarm optimization (PSO) -
Support Vector Regression (SVR)*

PERAMALAN KURS EUR/USD PADA *PASAR FOREX*
ONLINE MENGGUNAKAN *SUPPORT VECTOR*
REGRESSION YANG DIOPTIMASI DENGAN ALGORITMA
PARTICLE SWARM OPTIMIZATION

1. Bagaimana mendapatkan parameter optimal dari metode *support vector regression* dengan menggunakan algoritma *particle swarm optimization*.
2. Bagaimana peramalan kurs EUR/USD pada pasar *forex online* menggunakan *support vector regression* (SVR) dengan parameter yang telah optimal dan analisis tingkat eror dari hasil peramalan tersebut.

1. Data yang digunakan adalah data *closing* kurs EUR/USD yang bergerak setiap enam puluh menit yang diambil selama 20x24 jam.
2. Parameter yang dioptimasi adalah parameter C (penalti) dan σ (kernel *radial basis function*).
3. Perhitungan peramalan menggunakan Matlab.

1. Mendapatkan parameter yang optimal dari metode *support vector regression* (SVR) dengan menggunakan algoritma *particle swarm optimization* (PSO).
2. Meramalkan kurs EUR/USD pada pasar forex *online* menggunakan *support vector regression* (SVR) dengan parameter yang telah optimal dan menganalisis tingkat *error* dari hasil peramalan tersebut.

1. Sebagai alat bantu analisa bagi para pelaku pasar forex (trader forex) dalam mengambil keputusan beli dan jual.
2. Sebagai referensi dan informasi tentang penggunaan metode *Particle Swarm Optimization (PSO)* - *Support Vector Regression (SVR)*.

BAB II

Kajian Teori

Pasar Forex Online

Pasar forex online merupakan pasar dimana yang dijual adalah mata uang suatu negara terhadap mata uang negara lainnya yang melibatkan pasar-pasar uang utama di dunia selama 24 jam x 5 hari setiap minggu.



Peramalan

Peramalan adalah proses untuk memperkirakan berapa kebutuhan dimasa mendatang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan untuk memenuhi permintaan barang atau jasa.

Berdasarkan sifatnya peramalan terbagi menjadi dua yaitu :

1. metode kualitatif
2. metode kuantitatif

Peramalan Kurs EUR/USD Pada Pasar Forex Online

Peramalan kurs EUR/USD pada pasar *forex online* digunakan untuk memperkirakan harga kurs EUR/USD yang bergerak fluktuatif. Sehingga para trader dapat mengambil keputusan kapan untuk membeli dana kapan untuk menjual.



Support Vector Regression (SVR)

Merupakan metode pembelajaran mesin dimana SVR cabang dari *Support Vector Machine* (SVM) yang dapat menyelesaikan permasalahan regresi dan peramalan.

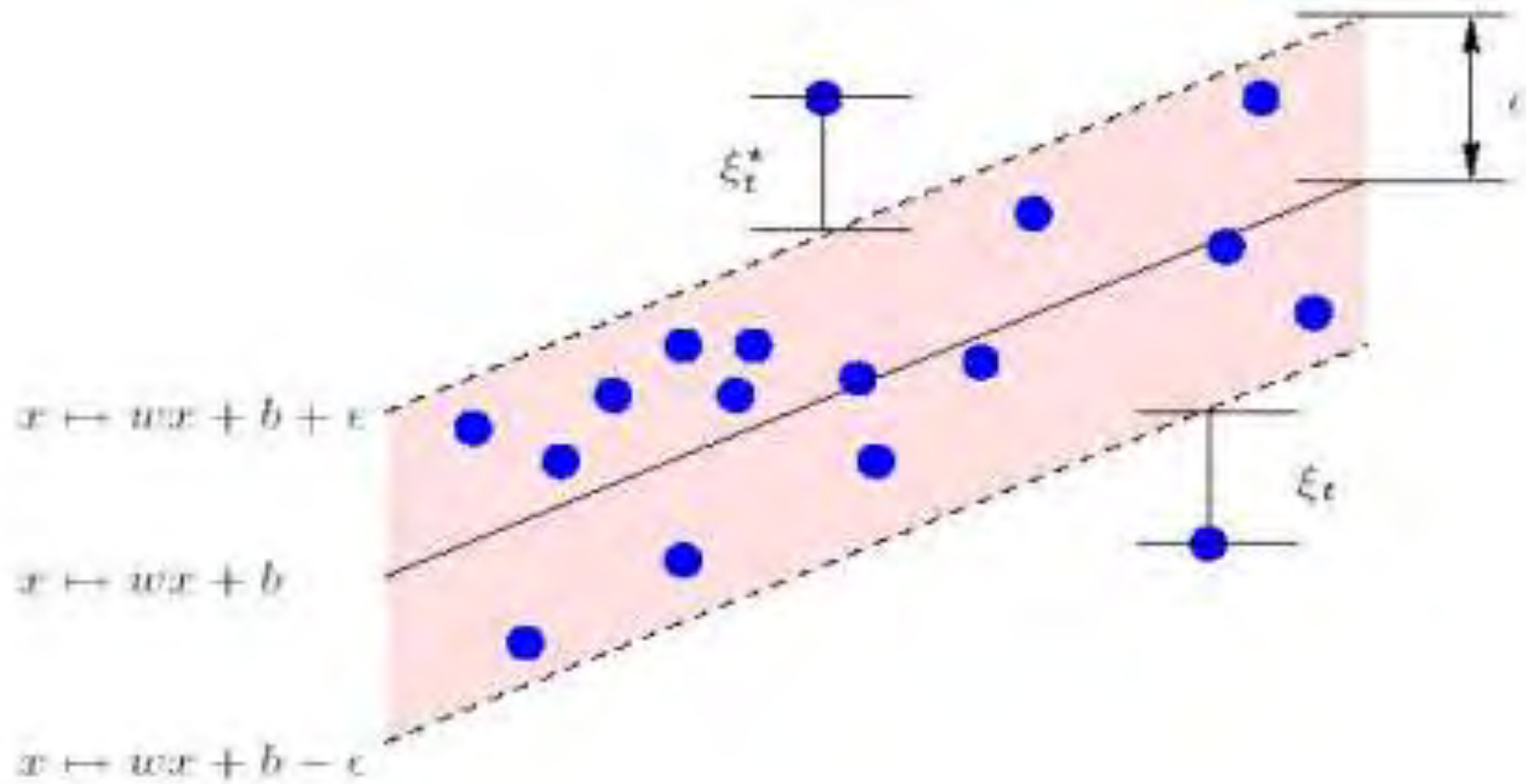


Misalkan diberikan l data pembelajaran $(x_1, y_1)(x_2, y_2), \dots, (x_l, y_l)$ $x_i \in \mathbb{R}^N$, x_i adalah data inputan atau vektor inputan, y_i adalah nilai output atau target.

Dengan SVR, kita hendak menemukan suatu fungsi $f(x)$ yang mempunyai deviasi (ε) paling besar dari target aktual y_i untuk semua data training.

Sehingga dengan SVR akan didapatkan suatu tabung seperti gambar berikut.

II. Kajian Teori



II. Kajian Teori

Model regresi mendefinisikan hubungan antara x_i dan $f(x_i)$ adalah sebagai berikut :

$$y = f(x_i) = (wx_i) + b$$

Dimana w adalah parameter bobot, b adalah parameter bias. Koefisien w dan b akan diestimasi dengan persamaan optimasi berikut :

$$\min_{w,b,\xi^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

dengan kendala :

$$((w \cdot x_i) + b) - y_i \leq \varepsilon + \xi_i, \quad i = 1, \dots, l$$

$$y_i - ((w \cdot x_i) + b) \leq \varepsilon + \xi_i, \quad i = 1, \dots, l$$

$$\xi_i^* \geq 0$$

C = koefisien pinalti; ξ = variable slack

I. Pendahuluan

Penyelesaian optimasi dalam bentuk primal pada persamaan (2) akan lebih mudah diselesaikan dengan mengubah bentuk primal ke bentuk dual sebagai berikut :

$$\max \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i') (\alpha_i - \alpha_i') K(x_i x_j) + \sum_{i=1}^l y_i ((\alpha_i - \alpha_i') - \sum_{i=1}^l (\alpha_i + \alpha_i') \xi_i^* \geq 0$$

Dengan kendala :

$$\sum_{i=1}^l (\alpha_i - \alpha_i') = 0$$
$$0 \leq \alpha_i \leq C, i = 1, \dots, l$$

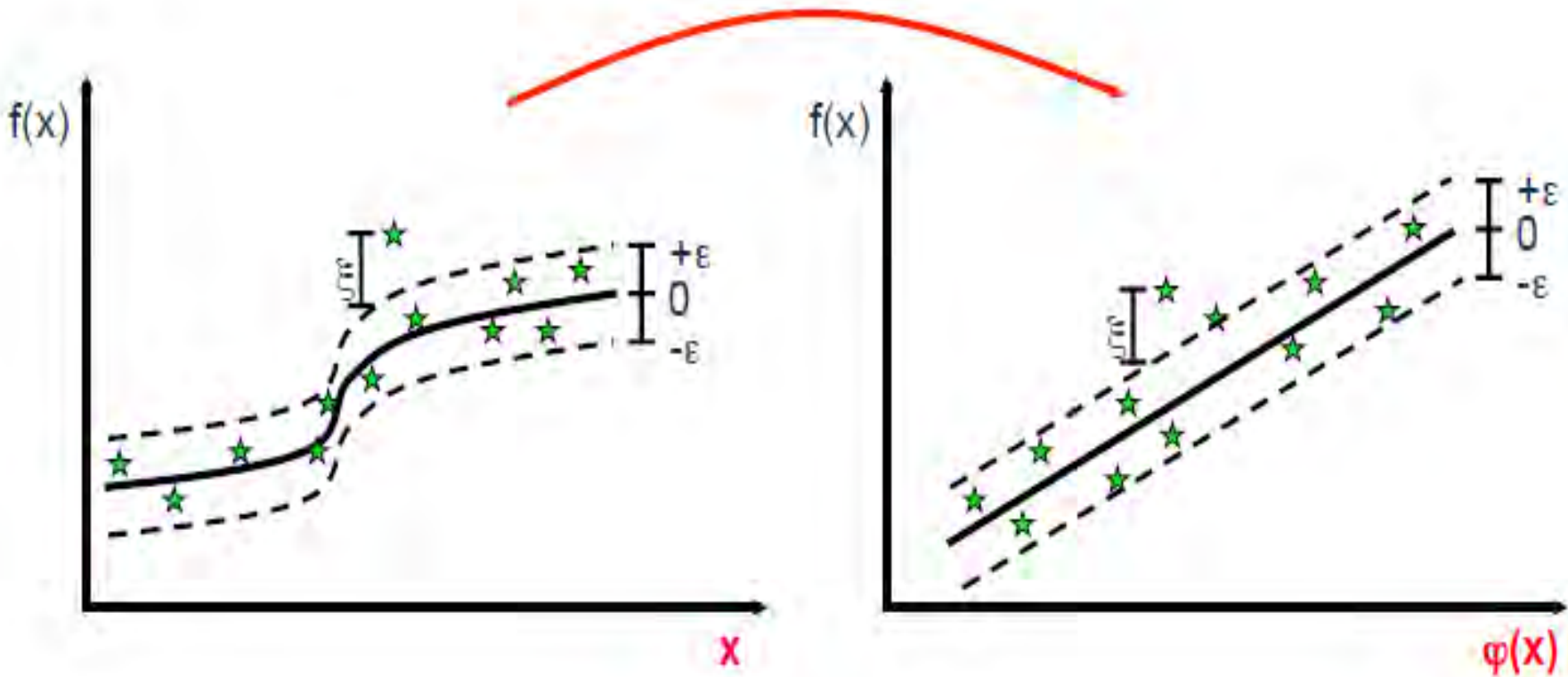
Untuk mengatasi permasalahan nonlinear maka kita gunakan metode kernel. $K(x_i x_j)$ adalah dot-product kernel yang didefinisikan sebagai $K(x_i x_j) = \phi^T(x_i) \phi(x_j)$. Sehingga fungsi regresi dirumuskan sebagai berikut :

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i') K(x_i x_j) + b$$

Metode Kernel

Metode kernel memberikan pendekatan alternatif dengan cara melakukan mapping data x dari input space ke feature space F melalui suatu fungsi ϕ sehingga $\phi: x \rightarrow \phi(x)$.

Dengan kernel, permasalahan nonlinear di input space bisa diselesaikan dengan membuat suatu fungsi pemisah yang linier pada feature space



Pemetaan kernel merubah permasalahan nonlinear menjadi linear

Penggunaan fungsi kernel yang berbeda akan menghasilkan model regresi yang berbeda pula. Fungsi kernel yang biasanya dipakai dalam literatur SVM :

1. Kernel linear : $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
2. Polynomial : $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)^p$
3. Kernel RBF : $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$

Pada penelitian ini fungsi kernel RBF digunakan untuk pemetaan kernel.

Didalam metode SVR terdapat parameter C (penalti) dan σ (RBF) yang dapat mempengaruhi hasil akhir dari peramalan.

Oleh karena itu untuk menghasilkan model peramalan yang baik maka, parameter-parameter tersebut di optimalkan menggunakan algoritma *particle swarm optimization*.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization merupakan metode optimasi yang pertama kali dikembangkan oleh James Kennedy dan Russell Eberhart. PSO merupakan metode optimasi dimana algoritmanya mengadopsi pada perilaku dan psikologi sosial sebuah organisme seperti rayap, lebah atau burung.

PSO dikembangkan dengan berdasarkan pada model berikut:

1. Ketika seekor burung mendekati target atau makanan (atau bisa minimum atau maximum suatu fungsi tujuan) secara cepat mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu
2. Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung
3. Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

II. Kajian Teori

- Dalam algoritma PSO, burung merupakan partikel.
- Setiap partikel diasumsikan memiliki dua karakteristik yaitu posisi dan kecepatan.
- Setiap partikel bergerak dalam *ruang/space* tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif.
- Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut.

Algoritma PSO

1. Inisialisasi parameter PSO, yaitu menentukan populasi kandidat solusi dari partikel untuk parameter σ (parameter kernel RBF) dan C (nilai pinalti)
2. Menentukan vektor posisi $X_j(i) = X_1(i), X_2(i), \dots, X_N(i)$ dan vektor kecepatan $V_j(i) = V_1(i), V_2(i), \dots, V_N(i)$, dimana $i = 0, 1, 2, \dots, \text{batas iterasi}$, $j = 1, 2, \dots, N$ dan $N = \text{banyaknya partikel}$.
3. Mendapatkan nilai kelayakan (*current fitness*) dari tiap-tiap kandidat solusi atau partikel dengan mengevaluasi nilai fungsi tujuan dimana pada penelitian ini fungsinya adalah meminimalkan nilai MAPE peramalan. Evaluasi nilai fungsi tujuan untuk setiap partikel pada $i = 0$ dan nyatakan dengan
$$f|X_1(0)|, f|X_2(0)|, \dots, f|X_N(0)|$$
4. Dari perhitungan fungsi objektif, dapat diketahui *local best fitness* (Pf_best) dan *local best position* (Px_best).
5. Mencari nilai *global best fitness* (Gf_best), yaitu nilai yang minimum dari *local best fitness* (Pf_best).

Algoritma PSO

6. Menentukan *global best position* (Gx_best), yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* (Px_best) dari partikel yang memenuhi persyaratan *global best fitness* (Gf_best).

7. Memperbarui kecepatan dengan menggunakan persamaan

$$V_j(i) = V_j(i - 1) + c_1 r_1 [P_{best,j} - x_j(i - 1)] + c_2 r_2 [G_{best} - x_j(i - 1)],$$
$$j = 1, 2, \dots, N$$

dimana c_1 dan c_2 masing-masing adalah *learning rates* untuk kemampuan individu (*cognitive*) dan pengaruh sosial (*group*), r_1 dan r_2 bilangan random yang berdistribusi uniform dalam interval 0 dan 1. Memperbarui posisi partikel selanjutnya dihitung dengan cara

$$X_j(i) = X_j(i - 1) + V_j(i); j = 1, 2, \dots, N$$

8. Ulangi langkah 2 sampai 7 hingga fungsi objektif konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi

BAB III

Analisis Data dan Pembahasan

A. Pengambilan Data

Data dari kurs *closing* EUR/USD pasar *forex online* diambil setiap satu jam dengan bantuan software trading Metatrader 4. Data historis diambil mulai pada tanggal 12 Januari 2014 sampai 16 Januari 2014 sehingga didapatkan 120 data.

B. Prapemrosesan Data



Normalisasi Data

Normalisasi dapat mempermudah perhitungan. Pada penelitian ini data dinormalisasi diantara rentang 0 - 1

Metode normalisasi *min-max* :

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

dengan :

- v'_i = data baru hasil normalisasi
- v_i = data inputan yang akan di normalisasi
- \min_A = nilai minimum dari data inputan
- \max_A = nilai maksimum dari data inputan
- new_min_A = nilai minimum dari data hasil normalisasi yang diinginkan
- new_max_A = nilai maksimum dari data hasil normalisasi yang diinginkan

Penyusunan Data

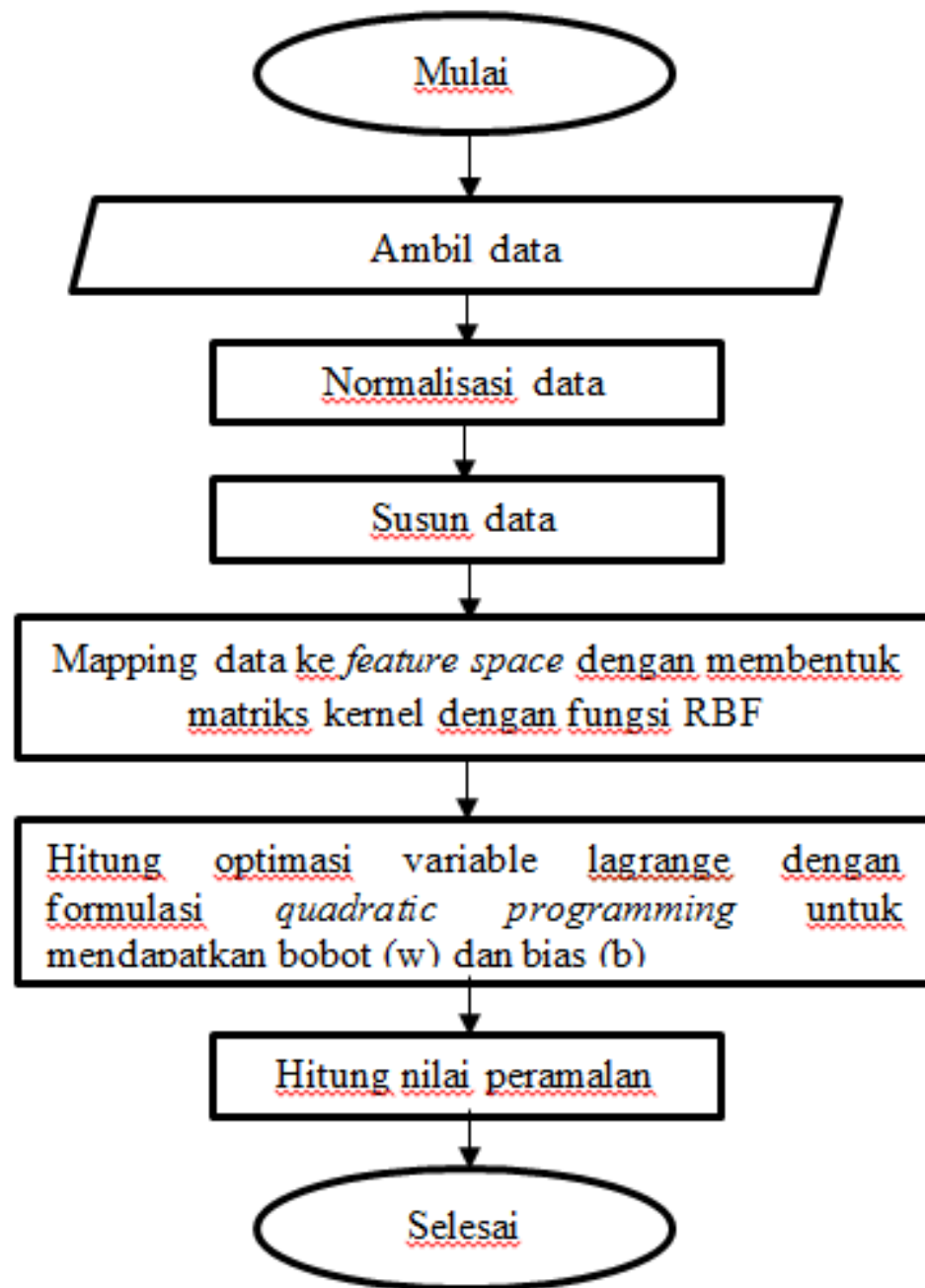
	f_1	f_2	f_3	f_4	<i>target</i>
Training	$data_1$	$data_2$	$data_3$	$data_4$	$data_5$
	$data_2$	$data_3$	$data_4$	$data_5$	$data_6$

	$data_{k-4}$	$data_{k-3}$	$data_{k-2}$	$data_{k-1}$	$data_k$
Testing	$data_{k-3}$	$data_{k-2}$	$data_{k-1}$	$data_k$	$data_{k+1}$
	$data_{k-2}$	$data_{k-1}$	$data_k$	$data_{k+1}$	$data_{k+2}$

	$data_{n-4}$	$data_{n-3}$	$data_{n-2}$	$data_{n-1}$	$data_n$

C. Training dan Testing SVR

Training data merupakan proses untuk mendapatkan nilai bobot (w) dan bias (b). Nilai dari w dan b akan dimasukkan ke model regresi.



Matriks Kernel

Fungsi kernel RBF : $= \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

$$\begin{bmatrix} K(x_1, x_1) & \dots & \dots & K(x_1, x_{93}) \\ \vdots & & & \\ K(x_1, x_{93}) & \dots & \dots & K(x_{93}, x_{93}) \end{bmatrix}$$

Perhitungan nilai w dan b

Menghitung nilai bobot dan bias dengan memasukkan nilai mapping dari fungsi kernel dan nilai C (koefisien pinalti) ke dalam quadratic programming (QP).

$$\max \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i') (\alpha_i - \alpha_i') K(x_i x_j) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i') - \sum_{i=1}^l (\alpha_i + \alpha_i') \xi_i^* \geq 0$$

Dengan kendala :

$$\sum_{i=1}^l (\alpha_i - \alpha_i') = 0$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, l$$

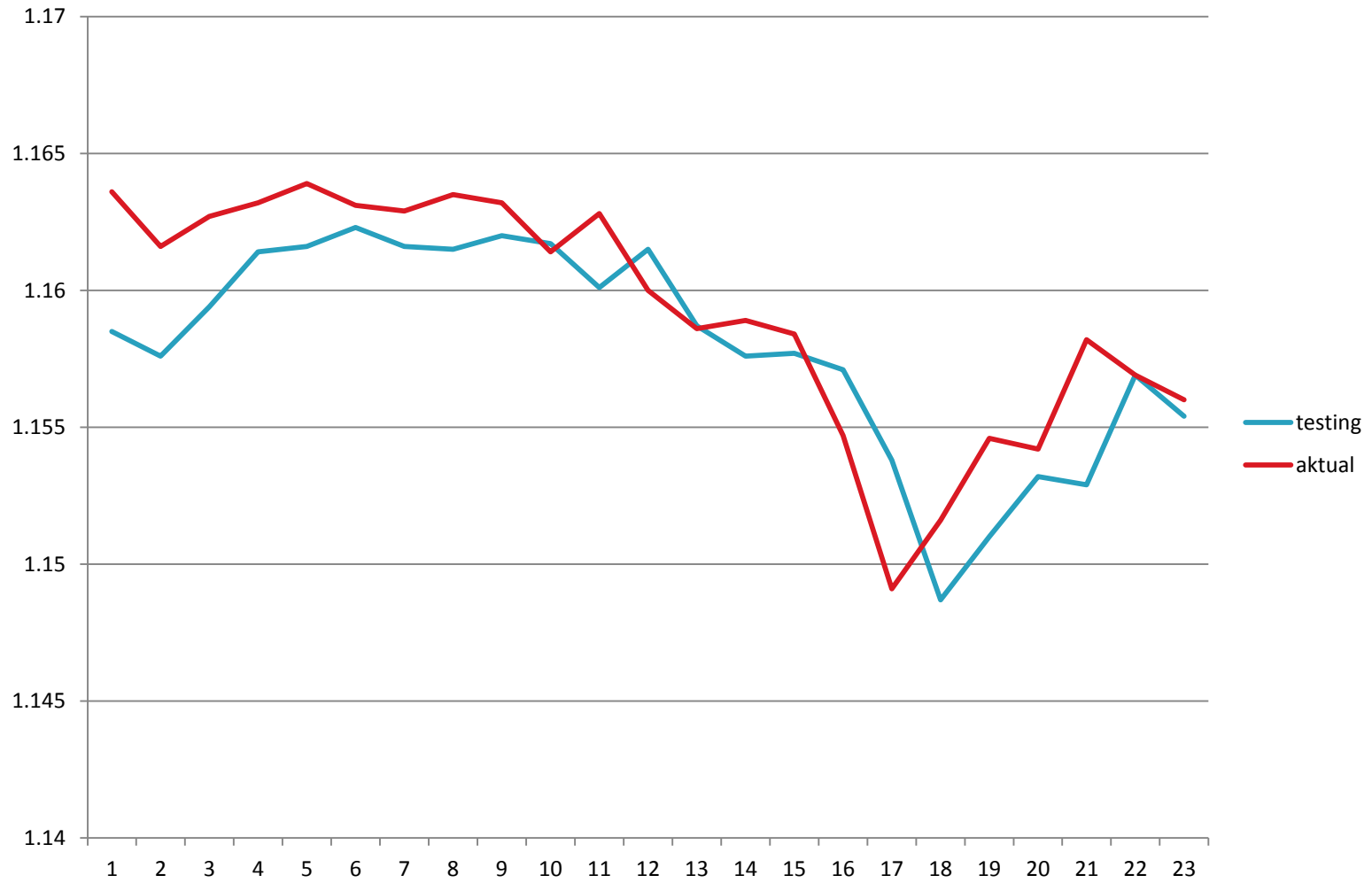
Dari perhitungan *quadratic programming* didapat nilai bobot (w) dan bias (b)

Testing

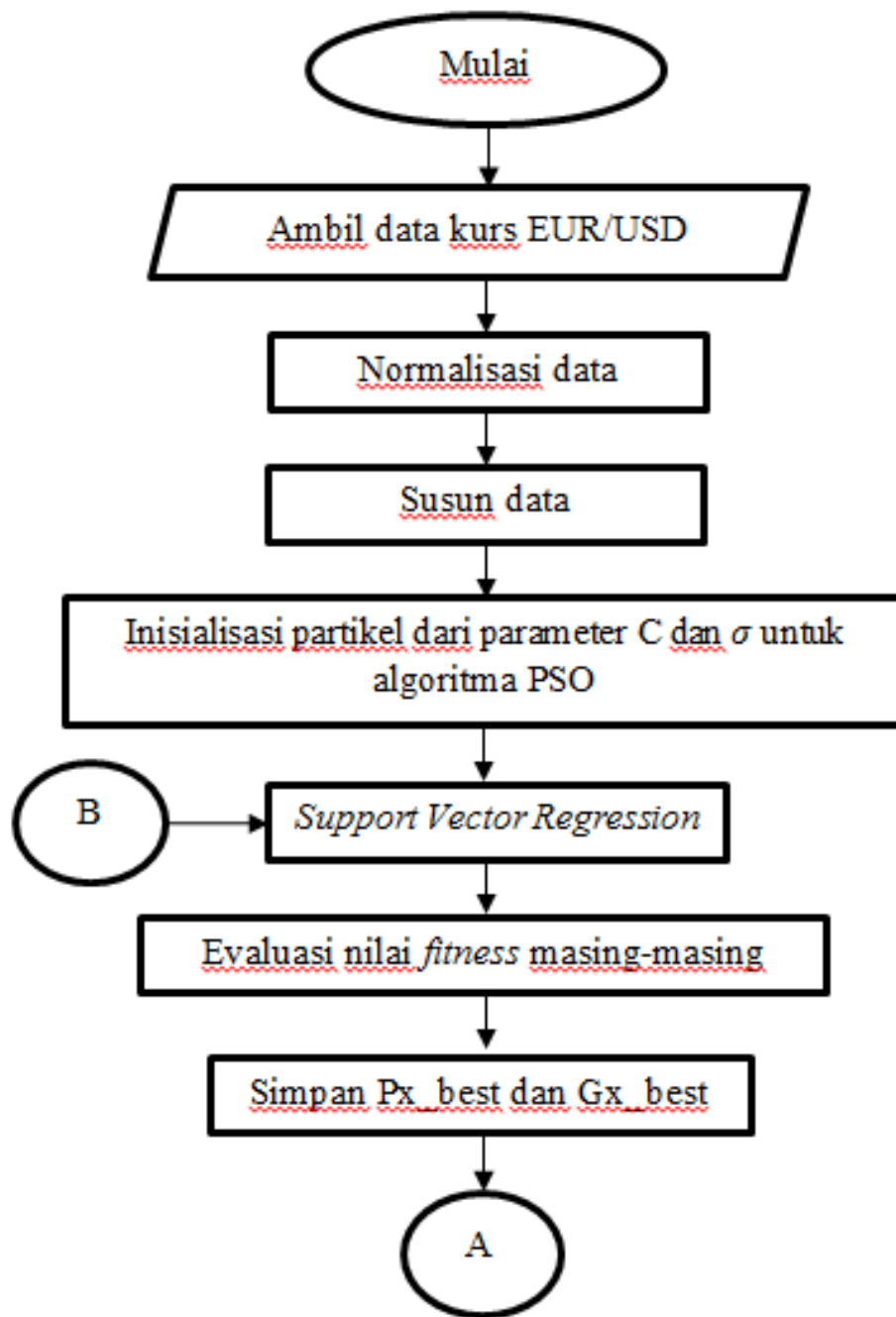
Setelah didapat nilai w dan b , maka didapatkan fungsi regresi sebagai berikut.

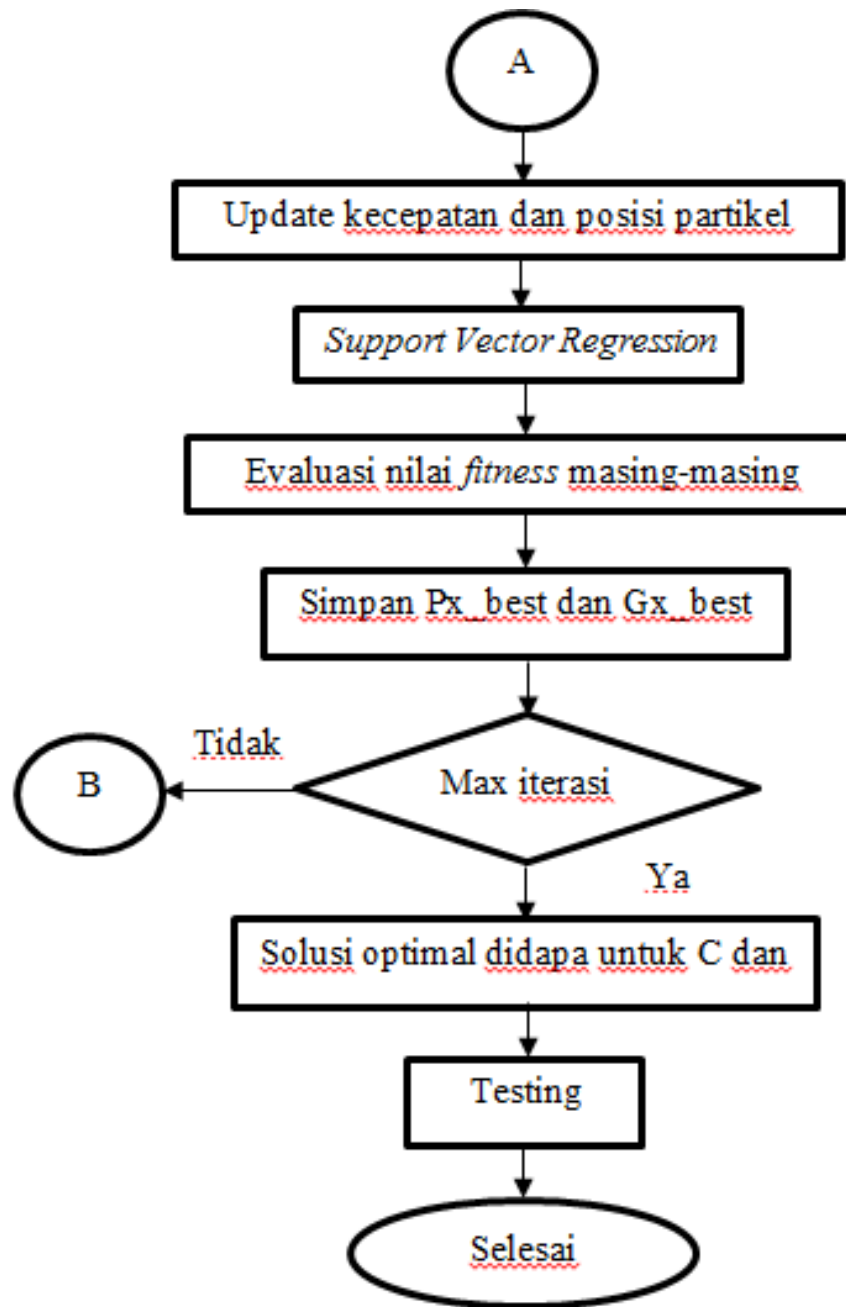
$$f(x) = (2.1979) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (7.1699) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \\ + (84.6638) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right)$$

Dihasilkan peramalan dengan nilai MAPE sebesar 0.2135%



D. Training dan Testing PSO-SVR





D. Training dan Testing PSO-SVR

1. Menentukan ukuran atau jumlah partikel (N) dari parameter PSO yaitu parameter C (*penalty cost*) dan σ (RBF). Pada penelitian ini partikel yang digunakan adalah sebanyak 20 partikel kemudian kita inisialisasi partikel tersebut dengan merandom nilai diantara 100-200 unuk parameter C dan 1-2 untuk parameter σ

partikel	c	σ
1	167.8735	1.6324
2	175.7740	1.0975
3	174.3132	1.2785
4	139.2227	1.5469
5	165.5478	1.9575
6	117.1187	1.9649
7	170.6046	1.1576
8	103.1833	1.9706
9	127.6923	1.9572
10	104.6171	1.4854
11	109.7132	1.8003
12	182.3458	1.1419
13	169.4829	1.4218
14	131.7099	1.9157
15	195.0222	1.7922
16	103.4446	1.9595
17	143.8744	1.6557
18	138.1558	1.0357
19	176.5517	1.8491
20	179.5200	1.9340

Training PSO-SVR

2. Dari perhitungan SVR didapat nilai bobot (w) dan bias (b). Setelah didapat nilai bobot (w) dan bias (b) kita testing nilai tersebut dengan menggunakan data training sebagai data testing. Dari proses ini didapatkan nilai ramalan kemudian kita hitung nilai MAPEnya dan dijadikan sebagai fungsi tujuan atau fungsi objektif dari algoritma PSO.

Perhitungan fungsi objektif digunakan untuk mendapatkan nilai kelayakan (*current fitness*) dari tiap-tiap kandidat solusi atau partikel.

3. Perhitungan fungsi objektif, dapat diketahui *local best fitness* (Pf_best) dan *local best position* (Px_best)

Partike I	Px_best C	Px_best σ	Pf_best C dan σ
1	167.8735	1.6324	3.5084
2	175.7740	1.0975	3.5124
3	174.3132	1.2785	3.5117
4	39.2227	1.5469	3.4909
5	165.5478	1.9575	3.5071
6	117.1187	1.9649	3.4777
7	170.6046	1.1576	3.5098
8	103.1833	1.9706	3.4675
9	127.6923	1.9572	3.4844
10	104.6171	1.4854	3.4686
11	109.7132	1.8003	3.4725
12	182.3458	1.1419	3.5156
13	169.4829	1.4218	3.5092
14	131.7099	1.9157	3.4868
15	195.0222	1.7922	3.5212
16	103.4446	1.9595	3.4677
17	143.8744	1.6557	3.4939
18	138.1558	1.0357	3.4903
19	176.5517	1.8491	1.8491
20	179.5200	1.9340	3.5143

Training PSO-SVR

4. Mencari nilai *global best fitness* (Gf_best), yaitu nilai yang minimum dari *local best fitness* (Pf_best). Dari perhitungan didapat nilai *global best fitness* (Gf_best) sebesar 3,6228.
5. Menentukan *global best position* (Gx_best), yaitu didapatkan dengan mengganti tiap kandidat solusi partikel dengan *local best position* (Px_best) dari partikel yang memenuhi persyaratan *global best fitness* (Gf_best). Sehingga didapatkan partikel yang menghasilkan nilai *global best fitness* (Gf_best) sebesar 3,6228 adalah pada saat $C = 104.9654$ dan $\sigma = 1.9106$

6. Memperbarui kecepatan dari partikel

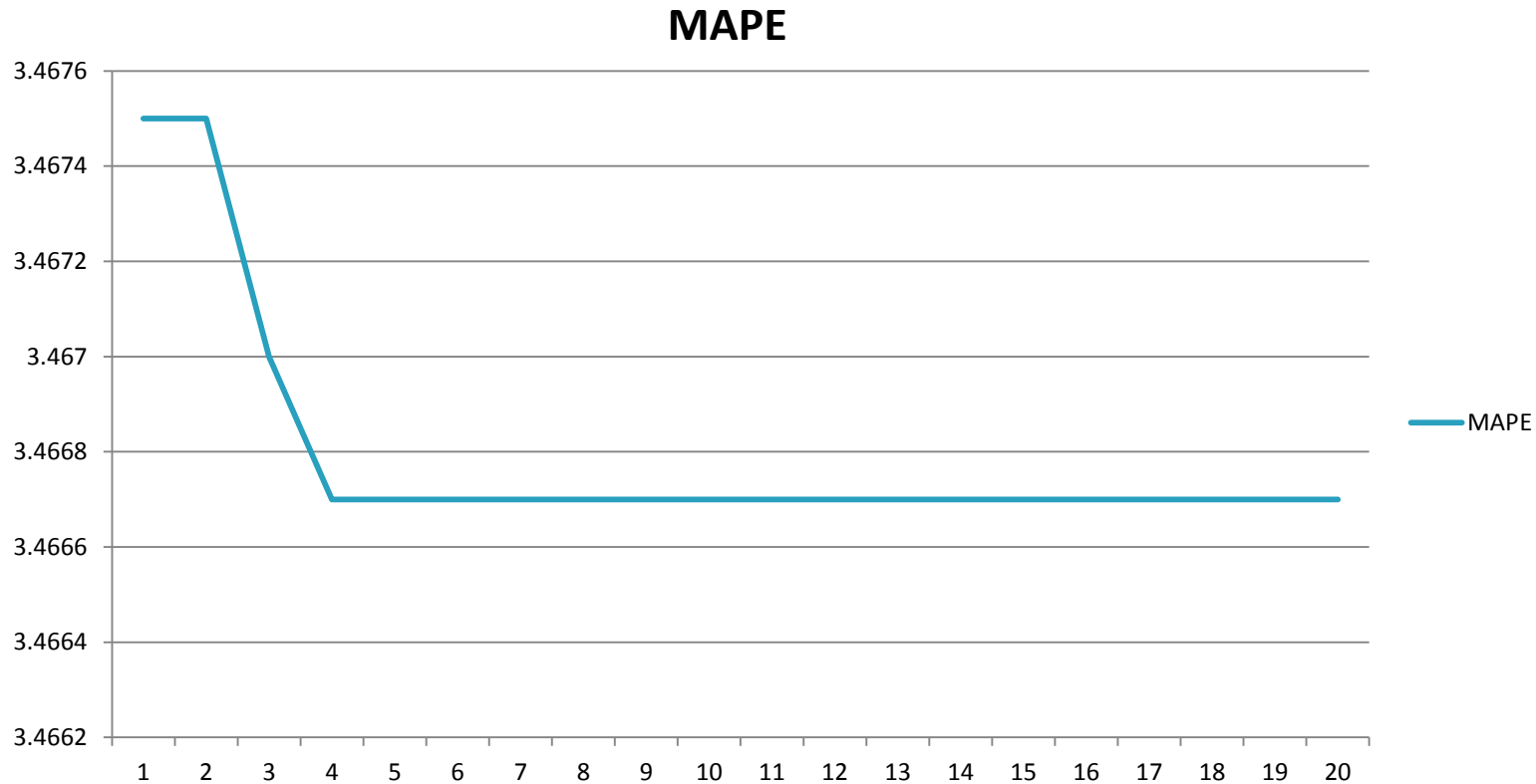
j	$V_j(1)c$	$V_j(1) \sigma$
1	-63.3659	0.3313
2	-71.1047	0.8552
3	-69.6738	0.6779
4	-35.3016	0.4150
5	-61.0878	0.0128
6	-13.6501	0.0056
7	-66.0411	0.7963
8	0	0
9	-24.0073	0.0132
10	-1.4045	0.4753
11	-6.3962	0.1668
12	-77.5419	0.8117
13	-64.9423	0.5376
14	-27.9427	0.0537
15	-89.9589	0.1747
16	-0.2560	0.0109
17	-39.8582	0.3084
18	-34.2566	0.9157
19	-71.8665	0.1190
20	-74.7740	0.0359

Posisi partikel yang telah diperbarui

j	$X_j(1) \text{ c}$	$X_j(1) \sigma$
1	104.5076	1.9637
2	104.6693	1.9527
3	104.6394	1.9564
4	103.9211	1.9619
5	104.4600	1.9703
6	103.4686	1.9705
7	104.5635	1.9540
8	103.1833	1.9706
9	103.6850	1.9703
10	103.2126	1.9607
11	103.3170	1.9671
12	104.8038	1.9536
13	104.5405	1.9594
14	103.7673	1.9695
15	105.0633	1.9669
16	103.1886	1.9704
17	104.0163	1.9641
18	103.8992	1.9515
19	104.6852	1.9681
20	104.7460	1.9698

Training PSO-SVR

7. Setelah didapat posisi partikel yang baru, hitung kembali fungsi tujuan dengan menggunakan posisi partikel sekarang. Jika nilai pf_best sekarang lebih baik dari nilai pf_best dari iterasi sebelumnya, maka nilai px_best sebelumnya digantikan dengan px_best sekarang
8. Ulangi langkah 3 sampai 7 hingga fungsi objektif atau nilai Gf_best konvergen ke suatu nilai atau hingga syarat batas iterasi terpenuhi.



Nilai MAPE mulai konvergen pada iterasi ke-6 dan pada iterasi ke-15 nilai MAPE= 3.4667 tidak berubah lagi. Sehingga didapatkan nilai parameter yang optimal $C = 16.9903$ dan $\sigma = 2.5464$.

Testing PSO-SVR

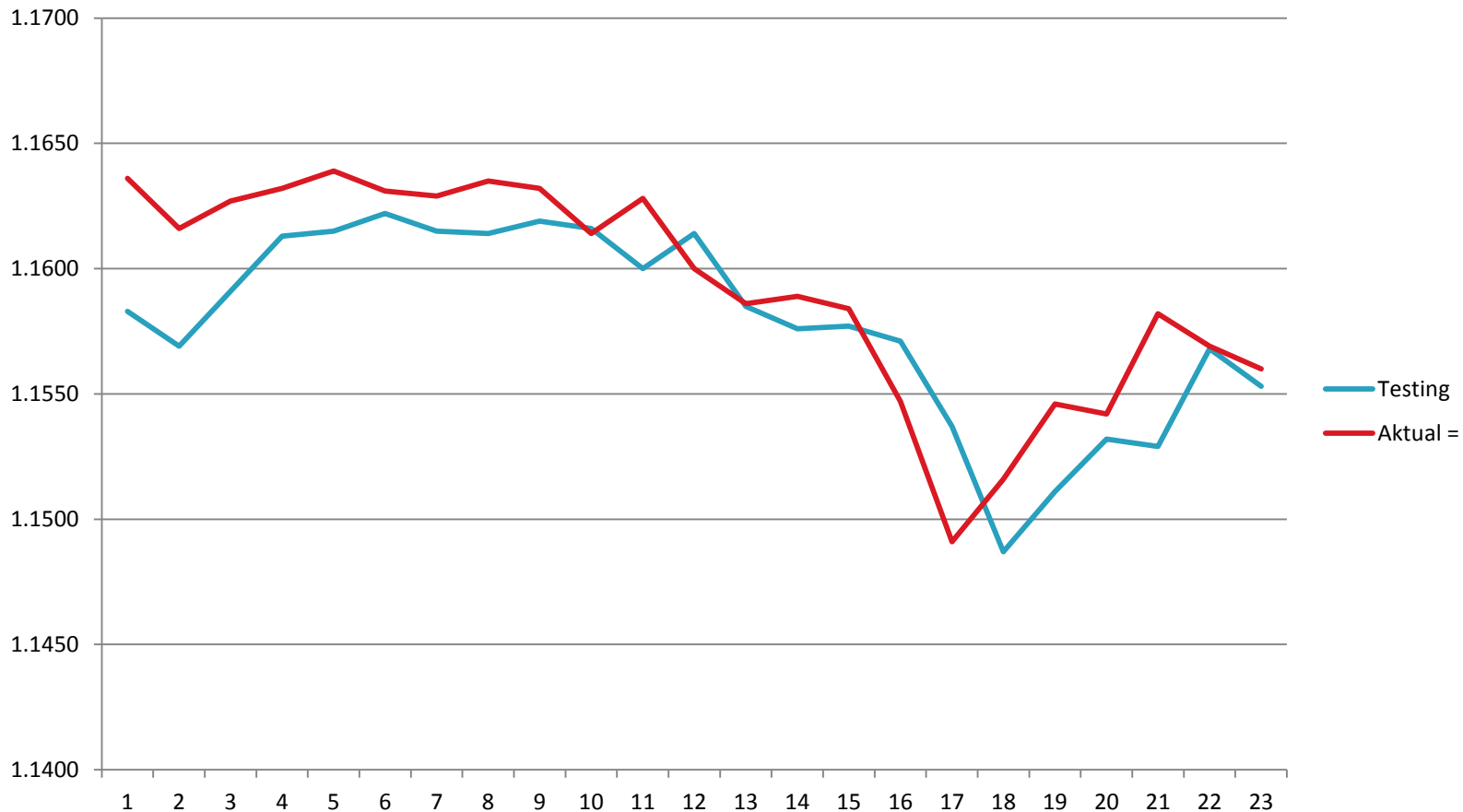
Dengan mendapatkan parameter C dan σ yang optimal dari SVR maka didapatkan pula W dan b yang optimal. Sehingga akan terbentuk model yang baik pada saat proses testing.

Untuk menguji seberapa baik model yang terbentuk maka dilakukan proses testing dengan menggunakan data testing yaitu periode 94-116.

Testing PSO-SVR

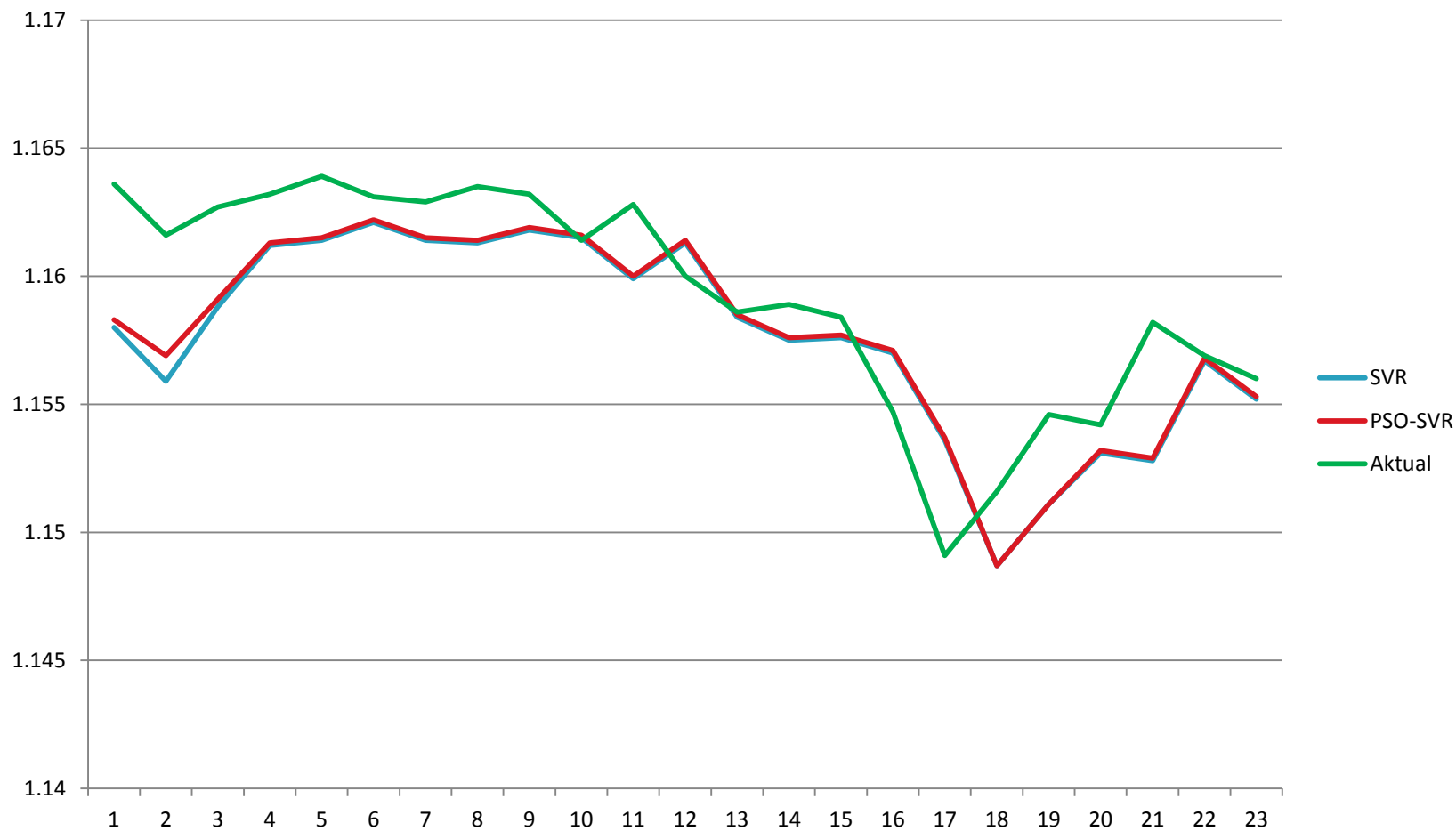
Model yang terbentuk pada saat proses testing SVR dengan parameter yang telah dioptimasi dengan algoritma PSO.

$$f(x) = \left((18.280) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + (64.017) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) + \dots \right. \\ \left. + (783.207) \left(\exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \right) \right) + 0$$



Perbandingan data aktual dengan hasil ramalan menggunakan metode SVR dengan parameter yang dioptimasi algoritma PSO. Testing menghasilkan nilai peramalan dengan MAPE sebesar 0.1895%

Perbandingan data aktual dengan peramalan yang dihasilkan metode SVR dan PSO-SVR



BAB IV

Kesimpulan

1. Parameter dari *support vector regression* yang telah dioptimasi dengan menggunakan *particle swarm optimization* optimal pada saat C bernilai 102.2522 dan σ bernilai 1.9724.
2. Tugas Akhir ini telah berhasil melakukan peramalan data kurs EUR/USD pada pasar forex *online* menggunakan metode peramalan *support vector regression* dengan nilai parameter $C = 110$ dan $\sigma = 1.5$ didapatkan MAPE sebesar 0.2135%. Untuk hasil peramalan menggunakan *support vector regression* dengan parameter yang telah optimal $C = 102.2522$ dan $\sigma = 1.9724$ didapatkan MAPE sebesar 0.1895%.

Daftar Pustaka

- [1] Anonim. 2009. “Struktur Investasi”. <http://belajarforex.com/sitting-duck/1-struktur-investasi.html>. Diakses pada tanggal 6 Februari 2015.
- [2] Kamruzzaman, J. dan Ruhul A Sarker. 2004. “Aplication of Support Vector Machine to Forex Monitoring”. IEEJ.
- [3] X. Liang, H. Zhang, J. Xiao, Y. Chen. 2009. “*Improving option price forecasts with neuralnetworks and support vector regressions*”. Neurocomputing 72 (13–15) 3055–3065.
- [4] Wang, X, Junhao Wen, Yihao Zhang, Yubiao Wang. 2013. “Real Estate Price Forecasting Based on SVM Optimized by PSO”. Optik, 125 (2014) 1439-1443.
- [5] J. Kennedy and R.C. Eberhart.1995. “A New Optimizer Using Particle Swarm Theory”.
- [6] Nasution, A. H. 1999. “Metode Aplikasi Peramalan (2 ed, Vol. 1)”. Binarupa Aksara
- [7] Santosa, B. 2007. “Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta: Graha Ilmu.
- [8] B. Scholkopf and A.J. Smola. 2002. “Learning with Kernels”. The MIT Press, Cambridge, Massachusetts.
- [9] Santosa, B. “Tutorial Particle Swarm Optimization”. Teknik Industri ITS. Surabaya.