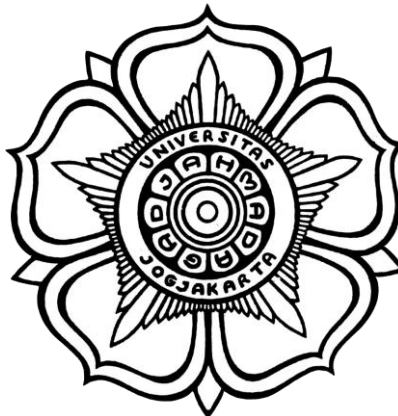


**SKRIPSI**

**PREDIKSI TIME SERIES  
MENGUNAKAN SUPPORT VECTOR MACHINE**

**TIME SERIES PREDICTION USING SUPPORT VECTOR MACHINE**



**WIDYA PERMANA KUSUMA  
04/177614/PA/09941**

**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

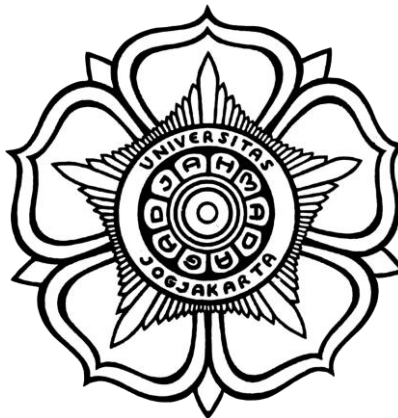
**2009**

**SKRIPSI**

**PREDIKSI TIME SERIES  
MENGUNAKAN SUPPORT VECTOR MACHINE**

**TIME SERIES PREDICTION USING SUPPORT VECTOR MACHINE**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat  
Sarjana Ilmu Komputer



**WIDYA PERMANA KUSUMA**  
04/177614/PA/09941

**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2009**

**HALAMAN PENGESAHAN**

**SKRIPSI**

**PREDIKSI TIME SERIES**

**MENGGUNAKAN SUPPORT VECTOR MACHINE**

Telah dipersiapkan dan disusun oleh

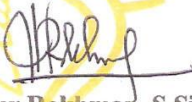
WIDYA PERMANA KUSUMA


04/177614/PA/09941

Telah dipertahankan di depan Tim Penguji  
pada tanggal 23 November 2009

Susunan Tim Penguji

  
**Drs. Janoe Hendarto, M.Kom**  
Pembimbing

  
**Nur Rokhman, S.Si., M.Kom**  
Penguji I

  
**Drs. Suprpto, M.Kom**  
Penguji II

  
**Agus Sihabuddin, S.Si., M.Kom**  
Penguji III

## **PERNYATAAN**

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 23 November 2009

Widya Permana Kusuma

## KATA PENGANTAR

Assalamualaikum Wr. Wb.

Segala puji dan syukur dipanjatkan ke hadirat Allah *Subhanahu Wata'ala*, karena hanya atas rahmat dan hidayah-Nya penulis dapat mencapai salah satu proses dari perjalanan panjang masa studi di Ilmu Komputer Universitas Gadjah Mada dengan selesainya skripsi yang berjudul “Prediksi Time Series Menggunakan Support Vector Machine”. Shalawat dan salam semoga selalu tercurah kepada Rasulullah, sahabat serta pengikutnya hingga akhir zaman.

Selama menjalani perkuliahan, banyak sekali dukungan moral maupun material, baik langsung ataupun tidak langsung dari berbagai pihak. Untuk itu, ucapan terima kasih penulis haturkan kepada

1. Bapak, Ibu, Kakak yang telah mendoakan, memberikan dukungan dan semangat sampai skripsi ini selesai.
2. Bapak Drs. Janoe Hendarto, M.Kom, selaku dosen pembimbing skripsi.
3. Bapak Drs. Medi, M.Kom, selaku dosen wali selama penulis menempuh studi di Ilmu Komputer Universitas Gadjah Mada.
4. Bapak Nur Rokhman, S.Si, M.Kom, Bapak Drs. Suprpto, M.Kom dan Bapak Agus Sihabuddin, S.Si, M.Kom selaku dosen penguji skripsi.
5. Seluruh dosen di lingkungan FMIPA UGM yang telah mendidik penulis selama menjadi mahasiswa.

6. Semua pihak yang telah berjasa kepada penulis yang tidak bisa disebutkan satu per satu.

Akhir kata, kesempurnaan adalah milik Allah dan kekurangan adalah milik hambaNya. Penulis berharap karya ini dapat bermanfaat.

Wassalamualaikum Wr. Wb.

Yogyakarta, November 2009

Penulis

## DAFTAR ISI

|  |            |
|--|------------|
| <b>HALAMAN JUDUL .....</b>                             | <b>i</b>   |
| <b>HALAMAN SAMPUL.....</b>                             | <b>ii</b>  |
| <b>HALAMAN PENGESAHAN .....</b>                        | <b>iii</b> |
| <b>HALAMAN PERNYATAAN.....</b>                         | <b>iv</b>  |
| <b>KATA PENGANTAR.....</b>                             | <b>v</b>   |
| <b>DAFTAR ISI.....</b>                                 | <b>vii</b> |
| <b>DAFTAR TABEL .....</b>                              | <b>x</b>   |
| <b>DAFTAR GAMBAR.....</b>                              | <b>xii</b> |
| <b>INTISARI .....</b>                                  | <b>xv</b>  |
| <b>ABSTRACT .....</b>                                  | <b>xvi</b> |
| <b>BAB I PENDAHULUAN.....</b>                          | <b>1</b>   |
| 1.1 Latar Belakang.....                                | 1          |
| 1.2 Rumusan Masalah.....                               | 3          |
| 1.3 Batasan Masalah .....                              | 3          |
| 1.4 Tujuan Penelitian .....                            | 4          |
| 1.5 Manfaat Penelitian .....                           | 4          |
| 1.6 Tinjauan Pustaka.....                              | 5          |
| 1.7 Metode Penelitian .....                            | 7          |
| 1.8 Sistematika Penulisan .....                        | 8          |
| <b>BAB II DASAR TEORI.....</b>                         | <b>10</b>  |
| 2.1 Peramalan .....                                    | 10         |
| 2.1.1 Definisi Peramalan .....                         | 10         |
| 2.1.2 Metode Peramalan.....                            | 12         |
| 2.2 Support Vector Machine .....                       | 16         |
| 2.2.1 <i>Support Vector Machine</i> untuk Permasalahan |            |

|  |           |
|--|-----------|
| Klasifikasi .....  | 23        |
| 2.2.1.1 Perhitungan <i>Support Vector Machine</i> untuk<br>Permasalahan Klasifikasi..... | 25        |
| 2.2.2 <i>Support Vector Machine</i> untuk Permasalahan Regresi...                        | 31        |
| 2.2.2.1 Perhitungan <i>Support Vector Machine</i> untuk<br>Permasalahan Regresi.....     | 32        |
| 2.2.3 Fungsi Kernel .....  | 36        |
| 2.3 Pemodelan Data Time Series .....   | 41        |
| 2.3.1 Pemilihan Feature.....   | 41        |
| 2.3.2 Segmentasi Data.....   | 42        |
| 2.3.3 Representasi Data.....   | 43        |
| 2.4 Pemilihan Parameter .....  | 45        |
| 2.5 Set Pelatihan, Set Pengujian dan Set Validasi .....                                  | 47        |
| 2.6 Teknik Validasi .....  | 48        |
| 2.7 Perhitungan Error .....  | 50        |
| 2.8 <i>Unified Modelling Language (UML)</i> .....  | 52        |
| 2.8.1 <i>Use Case Diagrams</i> .....   | 53        |
| 2.8.2 <i>Class Diagrams</i> .....  | 55        |
| 2.8.3 <i>Activity Diagrams</i> .....   | 57        |
| <b>BAB III ANALISIS DAN PERANCANGAN SISTEM.....</b>                                      | <b>59</b> |
| 3.1 Analisis Sistem.....   | 59        |
| 3.2 Spesifikasi Sistem .....   | 62        |
| 3.3 Rancangan Sistem .....   | 63        |
| 3.3.1 <i>Use Case Diagram</i> .....  | 63        |
| 3.3.2 <i>Activity Diagram</i> .....  | 67        |
| 3.3.3 <i>Class Diagram</i> .....   | 70        |



|   |            |
|---|------------|
| 3.3.4 Rancangan Antar Muka.....                               | 74         |
| 3.3.4.1 Form Training .....                                   | 74         |
| 3.3.4.2 Form Prediction.....                                  | 76         |
| 3.4 Rancangan Pengujian .....                                 | 77         |
| <b>BAB IV IMPLEMENTASI DAN PENGUJIAN .....</b>                | <b>83</b>  |
| 4.1 Pembangunan Sistem .....                                  | 83         |
| 4.2 Implementasi Algoritma.....                               | 84         |
| 4.2.1 <i>Class</i> svm_train .....                            | 84         |
| 4.2.2 <i>Class</i> svm_predict.....                           | 85         |
| 4.2.3 <i>Class</i> mainInterface .....                        | 87         |
| 4.2.3.1 Proses pelatihan .....                                | 87         |
| 4.2.3.2 Proses prediksi .....                                 | 90         |
| 4.3 Implementasi Aplikasi .....                               | 91         |
| 4.4 Pengujian.....  | 99         |
| 4.4.1. Data, Parameter dan Prosedur Pengujian .....           | 99         |
| 4.4.2. Pengujian .....  | 103        |
| 4.4.3 Analisis Pengujian .....                                | 111        |
| 4.4.3.1 Pengujian dengan Data Asli dan Data Skala .....       | 111        |
| 4.4.3.2 Pengujian dengan Data Asli<br>dan Data Prediksi ..... | 114        |
| <b>BAB V PENUTUP .....</b>                                    | <b>118</b> |
| 5.1 Kesimpulan .....  | 118        |
| 5.2 Saran.....  | 119        |
| <b>DAFTAR PUSTAKA .....</b>                                   | <b>120</b> |
| <b>LAMPIRAN.....</b>  | <b>126</b> |

## DAFTAR TABEL

|            |  |     |
|------------|--|-----|
| Tabel 2.1  | Kelebihan SVM dalam berbagai penelitian.....   | 22  |
| Tabel 2.2  | Fungsi Kernel .....  | 39  |
| Tabel 2.3  | Simbol dalam <i>Use Case Diagrams</i> .....  | 54  |
| Tabel 2.4  | Simbol dalam <i>Class Diagrams</i> .....   | 56  |
| Tabel 2.5  | Simbol dalam <i>Activity Diagrams</i> .....  | 58  |
| Tabel 4.1  | Parameter untuk data harga minyak .....  | 101 |
| Tabel 4.2  | Parameter untuk data nilai tukar Rupiah terhadap US Dollar .....                     | 102 |
| Tabel 4.3  | Kombinasi parameter terbaik data harga minyak .....                                  | 104 |
| Tabel 4.4  | Kombinasi parameter terbaik data nilai tukar Rupiah<br>Terhadap US Dollar .....      | 104 |
| Tabel 4.5  | Hasil prediksi data harga minyak untuk epsilon linear.....                           | 105 |
| Tabel 4.6  | Hasil prediksi data harga minyak untuk nu linear .....                               | 106 |
| Tabel 4.7  | MSE dan MAPE prediksi harga minyak .....   | 106 |
| Tabel 4.8  | Hasil prediksi data nilai tukar Rupiah untuk epsilon linear.....                     | 108 |
| Tabel 4.9  | Hasil prediksi data nilai tukar Rupiah untuk nu linear .....                         | 109 |
| Tabel 4.10 | MSE dan MAPE prediksi nilai tukar Rupiah .....                                       | 110 |
| Tabel 4.11 | Perbandingan prediksi data asli dan data skala<br>untuk harga minyak .....           | 112 |
| Tabel 4.12 | Perbandingan prediksi data asli dan data skala<br>untuk data nilai tukar Rupiah..... | 113 |
| Tabel 4.13 | Perbandingan prediksi data asli dan data prediksi<br>untuk data harga minyak.....    | 115 |

|  |     |
|--|-----|
| Tabel 4.14 Perbandingan prediksi data asli dan data prediksi |     |
| untuk data nilai tukar Rupiah.....                           | 116 |

## DAFTAR GAMBAR

|             |  |    |
|-------------|--|----|
| Gambar 2.1  | Konsep <i>Support Vector Machine</i> .....   | 17 |
| Gambar 2.2  | SVM berusaha menemukan bidang pemisah terbaik yang<br>memisahkan dua kelas -1 dan +1 .....   | 18 |
| Gambar 2.3  | Bidang pemisah .....   | 24 |
| Gambar 2.4  | <i>Support vector</i> dengan batas jarak maksimum.....   | 24 |
| Gambar 2.5  | Alternatif bidang pemisah (kiri) dan bidang pemisah terbaik<br>dengan margin (m) terbesar (kanan) .....  | 25 |
| Gambar 2.6  | Input SVM dan <i>feature space</i> .....   | 29 |
| Gambar 2.7  | Contoh transformasi untuk data yang tidak dapat<br>dipisahkan secara linear .....  | 30 |
| Gambar 2.8  | SVR bekerja dengan variabel C dan sembarang fungsi $\epsilon$ .<br>Variabel C digunakan untuk menghitung nilai error<br>di luar jangkauan $\epsilon$ . ..... | 32 |
| Gambar 2.9  | <i>Soft margin hyperplane</i> .....  | 33 |
| Gambar 2.10 | $\epsilon$ insensitive loss .....  | 34 |
| Gambar 2.11 | Fungsi Pemetaan .....  | 36 |
| Gambar 2.12 | Proses klasifikasi .....   | 47 |
| Gambar 2.13 | Metode <i>n-fold cross validation</i> .....  | 50 |
| Gambar 3.1  | Diagram alir sistem .....  | 60 |
| Gambar 3.2  | <i>Use Case Diagram</i> .....  | 64 |
| Gambar 3.3  | <i>Activity diagram</i> proses pelatihan .....   | 68 |

|             |   |    |
|-------------|---|----|
| Gambar 3.4  | <i>Activity diagram</i> proses prediksi .....   | 69 |
| Gambar 3.5  | <i>Class diagram</i> libsvm .....   | 71 |
| Gambar 3.6  | Form training.....  | 74 |
| Gambar 3.7  | Form prediction.....  | 76 |
| Gambar 3.8  | <i>Flowchart</i> pengujian SVReg.....   | 82 |
| Gambar 4.1  | <i>Source code</i> memanggil <i>method</i> svm_cross_validation .....                 | 84 |
| Gambar 4.2  | <i>Source code</i> menghitung MSE.....  | 85 |
| Gambar 4.3  | <i>Source code</i> membaca file input<br>dan diubah ke dalam bentuk svm_node .....    | 86 |
| Gambar 4.4  | <i>Source code</i> memanggil <i>method</i> svm_predict.....                           | 86 |
| Gambar 4.5  | <i>Source code</i> menghitung MSE dan MAPE.....                                       | 86 |
| Gambar 4.6  | <i>Source code</i> mengubah file input ke dalam svm_problem .....                     | 88 |
| Gambar 4.7  | <i>Source code</i> memanggil <i>method</i> svm_check_parameter .....                  | 89 |
| Gambar 4.8  | <i>Source code</i> memanggil <i>method</i> do_cross_validation<br>dan svm_train ..... | 89 |
| Gambar 4.9  | <i>Source code</i> membaca file input dan membuat file output.....                    | 90 |
| Gambar 4.10 | <i>Source code</i> memanggil <i>method</i> svm_load_model.....                        | 90 |
| Gambar 4.11 | <i>Source code</i> memanggil <i>method</i> svm_predict.....                           | 90 |
| Gambar 4.12 | <i>Source code</i> menampilkan nilai evaluasi<br>dan nilai prediksi .....             | 91 |
| Gambar 4.13 | Form training.....  | 92 |
| Gambar 4.14 | JDialog untuk memilih file data set training.....                                     | 92 |
| Gambar 4.15 | Form training sebelum sistem dijalankan .....   | 93 |

|             |  |    |
|-------------|--|----|
| Gambar 4.16 | Form training setelah sistem dijalankan.....         | 94 |
| Gambar 4.17 | Form training menyimpan model.....                   | 95 |
| Gambar 4.18 | Form prediction.....                                 | 96 |
| Gambar 4.19 | JDialog untuk memilih file data set prediction ..... | 97 |
| Gambar 4.20 | JDialog untuk memilih file model.....                | 97 |
| Gambar 4.21 | Form prediction sebelum sistem dijalankan.....       | 97 |
| Gambar 4.22 | Form prediction setelah sistem dijalankan.....       | 98 |

**INTISARI**  
**PREDIKSI TIME SERIES**  
**MENGGUNAKAN SUPPORT VECTOR MACHINE**

Oleh

Widya Permana Kusuma  
04/177614/PA/09941

*Support Vector Machine* adalah algoritma *learning machine* baru yang mendapat perhatian banyak peneliti sehubungan dengan prinsip *Structural Risk Minimization* yang mempunyai tingkat generalisasi yang lebih baik dan memiliki tingkat keberhasilan yang tinggi dalam *time series prediction*. Aplikasi SVM dalam *time series forecasting* dikenal dengan istilah *Support Vector Regression* (SVR). SVR merupakan salah satu pendekatan dalam menyelesaikan permasalahan regresi.

Tugas akhir ini akan membangun prototipe perangkat lunak yang menerapkan metode Support Vector Regression untuk melakukan prediksi terhadap data time series. Prototipe perangkat lunak diberi nama SVReg yang dibangun memanfaatkan library libsvm.

Pada tugas akhir ini dilakukan pengujian terhadap kinerja SVReg dengan menggunakan data harga minyak dan data nilai tukar Rupiah terhadap Dollar Amerika untuk mengetahui tingkat akurasi SVReg dalam melakukan prediksi. SVReg yang mengimplementasikan metode *Support Vector Machine* bisa memberikan kinerja yang sangat baik dalam melakukan prediksi yang ditunjukkan dengan nilai MAPE di bawah 3% untuk data dengan *noise* yang rendah.

Kata kunci : *Support Vector Machine, learning machine, Structural Risk Minimization, time series, prediksi*

**ABSTRACT**

**TIME SERIES PREDICTION  
USING SUPPORT VECTOR MACHINE**

By

Widya Permana Kusuma  
04/177614/PA/09941

Support Vector Machine (SVM) is a new machine learning algorithms that got the attention of many researchers in relation to the Structural Risk Minimization principle which has better generalization and has a high success rate in the time series prediction. SVM application in time series forecasting known as Support Vector Regression (SVR). SVR is one approach to solve the problem of regression.

This final task will build a software prototype that used Support Vector Regression method for prediction of time series data. The prototype software is named SVReg which used libsvm library.

In this final task, SVReg will be tested using oil prices data and Rupiah's exchange rate with US Dollar to determine the level accuracy prediction. SVReg that implemented Support Vector Machine method can give very good performance in the predictions indicated by MAPE value below 3% for data with low noise.

**Keyword** : Support Vector Machine, learning machine, Structural Risk Minimization, time series, prediction



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Peramalan mengenai kondisi yang terjadi di masa mendatang telah menjadi perhatian banyak pihak berkaitan dengan keuntungan yang dapat diambil dari pengetahuan ini. Berbagai macam metode dan teknik telah dikembangkan untuk mendapatkan tingkat akurasi yang semaksimal mungkin. Semakin tinggi tingkat akurasi semakin efektif pula tindakan yang bisa dilakukan oleh para pelaku ekonomi dalam menentukan berbagai keputusan.

Metode jaringan syaraf tiruan yang berkembang seiring berkembangnya kecerdasan buatan telah memberikan terobosan baru dalam melakukan peramalan. Algoritma *backpropagation* telah membuktikan keunggulannya dalam melakukan prediksi terhadap *Wall Street Journal's Dow Jones Industrial Index* (Darmadi, 1994) maupun dalam pemodelan *NASDAQ-GEM stock price relationship* (Ng et al., 2000). Akan tetapi karena dalam proses pembelajarannya jaringan syaraf tiruan hanya mencari bidang yang memisahkan saja, maka saat memperoleh bidang yang bisa memisahkan proses pelatihan akan berakhir. Karena itu yang sangat sering terjadi pada jaringan syaraf tiruan adalah *overfitting* (Kim dan Valdes, 2003). Proses ini baik sekali pada proses pelatihan tetapi tidak baik pada penggunaannya.

Untuk mengatasi masalah *overfitting*, Vapnik memperkenalkan *Support Vector Machine* (SVM). SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan bidang pemisah (*hyperplane*) terbaik yang memisahkan dua buah *class input space*. Prinsip ini berkaitan erat dengan *statistical learning theory* dan merepresentasikan metodologi terbaru dalam pengembangan jaringan syaraf tiruan (Vapnik, 1998 dan 1995) untuk aplikasi perhitungan fungsi, perhitungan regresi dan *signal processing* (Vapnik et al., 1996). Termasuk juga pengenalan pola (Burgess, 1998), *time series forecasting* (Cao, 2003), dan prediksi (Muller et al., 1997). Dengan adanya Vapnik's  $\epsilon$ -insensitive loss function SVM bisa menyelesaikan regresi non linear dan permasalahan *time series* dengan kinerja yang sangat baik (Muller et al., 1997).

Berbeda dengan strategi jaringan syaraf tiruan yang berusaha mencari bidang pemisah antar kelas, SVM berusaha menemukan bidang pemisah yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier* dan selanjutnya dikembangkan agar dapat bekerja pada problem non linear dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Perkembangan ini memberikan rangsangan minat penelitian di bidang pengenalan pola untuk investigasi potensi kemampuan SVM secara teoritis maupun dari segi aplikasi. Dewasa ini SVM telah berhasil diaplikasikan dalam problema dunia nyata dan secara umum memberikan solusi yang lebih baik dibandingkan metode konvensional seperti misalnya *Artificial Neural Network*.

Penelitian kali ini akan memanfaatkan *Support Vector Machine* dalam melakukan prediksi data *time series* serta menguji tingkat akurasi dan kinerjanya menggunakan *Mean Squared Error* dan *Mean Absolute Percentage Error*.

## 1.2 Rumusan Masalah

Berdasarkan uraian di atas, maka dapat dirumuskan masalah-masalah sebagai berikut.

1. Mengembangkan aplikasi prediksi menggunakan metode *Support Vector Machine*.
2. Melakukan prediksi terhadap data pada waktu tertentu untuk kemudian dibandingkan dengan data sebenarnya untuk memperoleh nilai error prediksi.

## 1.3 Batasan Masalah

Untuk memenuhi tujuan penelitian ini dan mencegah agar permasalahan tidak terlalu meluas, diperlukan adanya batasan masalah. Adapun batasan permasalahan yang akan diteliti adalah sebagai berikut.

1. Peramalan dilakukan dengan menggunakan teknik *Support Vector Regression* berdasarkan metode *Support Vector Machine*.
2. Hasil peramalan dengan menggunakan *Support Vector Machine* akan dibandingkan dengan hasil sebenarnya dengan menggunakan *Mean Square Error* (MSE) dan *Mean Absolute Percentage Error* (MAPE).

MSE digunakan dalam evaluasi pembentukan model sedangkan MAPE digunakan sebagai evaluasi nilai prediksi.

3. Peramalan menggunakan data *time series*.

Data yang akan digunakan adalah data harga minyak mentah dunia berdasarkan WTI dan nilai tukar Rupiah terhadap US Dollar. WTI sendiri merupakan kependekan dari West Texas Intermediate yang juga dikenal sebagai Texas Light Sweet merupakan *benchmark* harga minyak mentah dunia.

4. Perangkat lunak hasil tugas akhir yang mengimplementasikan metode *Support Vector Machine* dibangun menggunakan bahasa pemrograman Java dan *library* libsvm.

#### 1.4 Tujuan Penelitian

Beberapa macam studi mengenai teknik peramalan banyak dilakukan oleh berbagai pihak hingga saat ini, terutama mengenai penggunaan teknik-teknik terbaru untuk meningkatkan kinerja dan ketepatan peramalan. Oleh karena itu, tujuan dilaksanakannya penelitian ini adalah sebagai berikut.

1. Mengimplementasikan *Support Vector Machine* dalam melakukan peramalan.
2. Melakukan perbandingan ketepatan hasil peramalan dengan menggunakan metode *Support Vector Machine* dengan hasil sebenarnya.

### 1.5 Manfaat Penelitian

Manfaat yang diharapkan dicapai dari penelitian ini adalah sebagai berikut.

1. Meningkatkan studi mengenai *Support Vector Machine*, terutama dalam penggunaannya sebagai metode peramalan.
2. Mengetahui seberapa tepat perkiraan yang bisa dibuat dengan menggunakan metode *Support Vector Machine*.

### 1.6 Tinjauan Pustaka

Pada awalnya, *Support Vector Machine* (SVM) didesain untuk menyelesaikan permasalahan pengenalan pola dengan menentukan suatu bidang pemisah terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space* dengan melakukan optimasi terhadap jarak di antaranya. Konsep ini didasarkan pada metode *Structural Risk Minimization* dan teori *statistical learning* dimana tingkat kesalahan pada data yang digunakan untuk tes dibatasi oleh tingkat kesalahan pada proses pelatihan dan dibatasi oleh *Vapnik-Chervonenkis dimension* (Palit dan Popovic, 2005).

Akan tetapi seiring berkembangnya penelitian di bidang komputasi dan statistik, SVM telah dikembangkan menjadi suatu metode untuk melakukan peramalan di berbagai bidang. Lung (2006) menuliskan keberhasilan SVM dalam *time series prediction* telah terbukti pada beberapa penelitian *Stock Price Forecasting* (Bao et al., 2004), *Traffic Speed Prediction* (Vanajakshi dan Rilett, 2004) dan *Travel Time Series Prediction* (Wu et al., 2004).

Muller et al. (2000) menyarankan pemilihan parameter *Support Vector Machine* yang tepat agar model SVM mampu menangani permasalahan *noise* pada data yang cukup tinggi dan regularisasi.

Chen et al. (2004) menggunakan *Support Vector Machine* untuk melakukan *electricity load forecasting* dengan menggunakan segmentasi data yang tepat untuk melakukan peramalan. Penelitian ini berhasil memenangi kompetisi yang diselenggarakan oleh EUNITE mengungguli penggunaan metode peramalan lain (Rojas dan Pomares, 2004).

Lung (2006) dalam penelitiannya menggunakan *Support Vector Machine* dengan polynomial kernel untuk melakukan peramalan saham *Kuala Lumpur Stock Exchange* (KLSE). Kinerja SVM dibandingkan dengan *Back Propagation Neural Network* (BPNN) dan hasilnya menunjukkan bahwa BPNN menunjukkan inkonsistensi dalam proses prediksi sementara SVM menghasilkan kinerja yang terjaga yang ditunjukkan dengan nilai MSE yang lebih kecil. Lung (2006) menyatakan bahwa SVM dapat dijadikan alternatif yang menjanjikan untuk melakukan *financial time series forecasting*.

Xie et al. (2006) menyarankan penggunaan empat tahapan dalam melakukan prediksi *Support Vector Machine*, yakni *data sampling*, *sample preprocessing*, *training and learning*, dan *out of sample forecasting*. Dengan menggunakan empat tahapan ini, hasil prediksi SVM mampu mengungguli metode ARIMA dan BPNN dalam melakukan peramalan.

## 1.7 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Studi Literatur

Studi literatur mengenai *Support Vector Machine* dilakukan melalui media cetak, seperti buku, paper, dan tugas akhir, serta media elektronik dengan mengakses situs-situs yang relevan.

2. Analisis dan Desain Sistem

Pada tahapan ini dilakukan perancangan aplikasi secara detail, mencakup semua spesifikasi yang akan dibangun, serta pemilihan *tools* dan *platform* yang akan digunakan.

3. Implementasi

Pada tahapan ini dilakukan pembangunan aplikasi berdasarkan spesifikasi yang sudah dirancang pada tahap sebelumnya, termasuk instalasi *tools* yang dibutuhkan dan kegiatan *coding*.

4. Pengujian dan Evaluasi

Melakukan pengujian dan evaluasi terhadap aplikasi yang dibangun untuk mengetahui berbagai kelebihan dan kekurangannya.

## **1.8 Sistematika Penulisan**

Untuk mempermudah mendapat gambaran tentang isi dari pembahasan tugas akhir ini, penulis menyusun secara sistematis hal tersebut dalam beberapa bab berikut ini :

### **BAB I     PENDAHULUAN**

Bab ini menjelaskan latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, tinjauan pustaka yang digunakan sebagai bahan referensi dalam penulisan tugas akhir, metode penelitian yang digunakan, dan sistematika penulisan tugas akhir.

### **BAB II    DASAR TEORI**

Bab ini membahas teori-teori, metode, dan konsep-konsep yang menjadi landasan topik yang dibahas pada tugas akhir ini, yaitu mengenai peramalan dan *Support Vector Machine* yang digunakan pada penelitian tugas akhir ini.

### **BAB III   ANALISIS DAN DESAIN**

Bab ini berisi rancangan dan desain sistem yang akan dibangun.

### **BAB IV   IMPLEMENTASI DAN PEMBAHASAN**

Bab ini membahas implementasi dari sistem yang dibangun dan cara kerja sistem secara lebih detail. Pada bab ini juga dibahas mengenai hasil peramalan yang didapatkan dari penggunaan sistem.



## BAB V    PENUTUP

Bab ini berisi kesimpulan dari keseluruhan tugas akhir dan saran-saran yang dapat dijadikan bahan masukan yang berarti untuk pengembangan sistem lebih lanjut.

## BAB II

### DASAR TEORI

#### 2.1 Peramalan

##### 2.1.1 Definisi Peramalan

Asim et al. menyarankan model *time series* untuk melakukan peramalan dibandingkan dengan pendekatan tradisional (Lung, 2006). Bowerman dan O'Connell mendeskripsikan peramalan adalah kegiatan membuat prediksi mengenai kejadian di masa mendatang sementara data *time series* digunakan dalam membuat prediksi (Lung, 2006). Ini berarti data *time series* bisa dimanfaatkan untuk membuat prediksi kondisi atau kejadian di masa mendatang.

Peramalan *time series* menggunakan data  $x_{t-n}, \dots, x_{t-2}, x_{t-1}, x_t$  untuk memprediksi nilai  $x_{t+1}, x_{t+2}, \dots$ . Tujuannya adalah untuk mengamati dan membuat model *time series* yang bisa digunakan untuk mengetahui nilai di masa mendatang secara akurat.

Fayyad et al. menjelaskan prediksi yang juga dikenal sebagai regresi adalah fungsi pembelajaran yang memetakan item data ke dalam variabel prediksi *real-valued* (Lung, 2006). Regresi adalah studi mengenai hubungan di antara variabel-variabel tersebut, tujuan penting mengenai apa yang diprediksi, atau perhitungan nilai dari satu variabel berdasarkan nilai dari variabel lain yang mempunyai kaitan dengan variabel yang akan diprediksi. Setelah pola dibuat, prediksi bisa dilakukan. Di

sampling itu, prediksi juga berhubungan dengan penemuan dan penggunaan pola dalam jumlah data *time series* yang cukup besar dimana prediksi mengenai nilai yang akan datang bisa dibuat (Lung, 2006).

Dalam Wibowo (2003), McLeod menyatakan ada beberapa fakta dasar dalam proses prediksi, yakni:

1. Prediksi adalah proyeksi dari masa lalu sehingga dasar terbaik untuk memperkirakan apa yang terjadi di masa depan adalah dengan melihat masa lalu. Inilah alasan mengapa data sangat penting dalam melakukan prediksi karena data ini akan memberikan data hipotesis.
2. Semua prediksi terdiri dari keputusan semi terstruktur. Keputusan tersebut didasarkan oleh beberapa variabel yang dapat diukur dengan mudah dan beberapa tidak dapat diukur.
3. Tidak ada teknik prediksi yang sempurna.

Peramalan *financial time series* dilakukan berdasarkan keteraturan data yang dilakukan melalui observasi. Akan tetapi dikarenakan pada data *time series* kadang terdapat *noise* dan tidak linear serta tidak mempunyai trend yang tetap, beberapa *financial time series* dengan pendekatan tradisional kadang tidak bisa diprediksi. Oleh karena itu dibutuhkan suatu algoritma pembelajaran yang bisa menangani karakteristik data *time series* ini. Beberapa algoritma *learning machine* baru diperkenalkan dalam beberapa tahun belakangan.

### 2.1.2 Metode Peramalan

Pendekatan analisis *time series* yang klasik (1970) menggunakan algoritma *Box Jenkins* untuk mengubah data dari proses stasioner ke dalam bentuk *Auto Regression* dan komponen *Moving Average* (ARMA). Box dan Jenkins menjelaskan metode ini memungkinkan perhitungan *actual pulses* atau *inclusion of forecast pulses* yang tidak sesuai dengan metodologi ARMA (Lung, 2006).

Dalam perkembangan dunia peramalan, muncul metode *Artificial Neural Network* (ANN). ANN adalah pendekatan non linear dan aliran data yang tergantung pada data yang akan "dipelajari", tanpa hipotesis a priori mengenai bentuk hubungan data, yang membuatnya sesuai untuk data yang kompleks. Pendekatan ini menggunakan beberapa *layer node* yang masing-masing terhubung satu sama lain dengan bobot tertentu. Dengan menambahkan *layer* dan memori, input sebelumnya disimpan dan output dari jaringan syaraf tiruan bisa membentuk model *time series*. Akan tetapi jaringan syaraf tiruan sangat bergantung pada struktur jaringan dan kompleksitas sampel yang dapat mengakibatkan *overfitting* dan generalisasi yang rendah (Kim dan Valdes, 2003; Zhu et al., 2002).

*Wavelet networks* diusulkan oleh Zhang dan Benveniste sebagai alternatif bagi *feedforward neural network* (Alexandridisi dan Livanis, 2008). *Wavelet networks* adalah jaringan syaraf tiruan dengan satu *hidden layer* yang menggunakan *wavelet* sebagai fungsi aktivasi, bukan fungsi klasik sigmoid. Gao dan Tsoukalas (2001) mempertimbangkan penggunaan *wavelet networks* sebagai salah satu metode yang menjanjikan dalam menyelesaikan permasalahan *electricity load forecasting*.

Alexandridisi dan Livanis (2008) menggunakan *wavelet networks* untuk menyelesaikan permasalahan *crude oil forecasting*. *Wavelet networks* ini mempunyai kinerja yang lebih baik daripada pendekatan jaringan syaraf tiruan yang lain dalam menangani permasalahan *overfitting* dan data yang *noisy* (Wong dan Wong, 2004).

Untuk mengatasi masalah *overfitting* yang dihadapi oleh metode jaringan syaraf tiruan, Vapnik memperkenalkan *Support Vector Machine* (SVM). *Support Vector Machine* adalah algoritma *learning machine* baru yang mendapat perhatian banyak peneliti sehubungan dengan prinsip *Structural Risk Minimization* yang mempunyai tingkat generalisasi yang lebih baik dan memiliki tingkat keberhasilan yang tinggi dalam *time series prediction* (Johan, 2001). SVM bekerja dengan menemukan bidang pemisah yang memisahkan set pelatihan dalam *feature space* yang dibentuk dari fungsi kernel sebagai *inner product* dalam algoritma tersebut (Cristianini et al., 2000). Lung (2006) menuliskan keberhasilan SVM dalam *time series prediction* telah terbukti pada beberapa penelitian *Stock Price Forecasting* (Bao et al., 2004), *Traffic Speed Prediction* (Vanajakshi dan Rilett, 2004) dan *Travel Time Series Prediction* (Wu et al., 2004).

Dalam beberapa tahun terakhir, Wang et al. (2004) mengusulkan metodologi terbaru untuk menangani prediksi *time series* yang kompleks dengan memperkenalkan metode TEI@I. Metode ini mengintegrasikan *text mining*, ekonometrika dan kecerdasan buatan. Dalam metode ini model ekonometrika digunakan untuk menangani komponen linear pada metode *time series* (seperti trend umum) sedangkan komponen non linear pada data *time series* (seperti *error terms*)

dimodelkan dengan menggunakan model kecerdasan buatan. Di samping itu pengaruh peristiwa yang terjadi dalam dunia nyata yang terjadi di masa mendatang yang tidak terduga dan tidak teratur dieksplorasi dengan menggunakan *web-based text mining* (WTM) dan teknik *rule based expert system* (RES). Dalam penelitiannya mengenai *Crude Oil Forecasting*, Wang et al. (2004) menyatakan bahwa TEI@I memberikan performa yang baik dalam *crude oil forecasting* dengan menggunakan metode *Back Propagation Neural Network* (BPNN) sebagai model kecerdasan buatan yang terintegrasi pada metode TEI@I.

Akan tetapi Xie et al. (2006) menyatakan bahwa BPNN yang digunakan pada metode TEI@I mempunyai kekurangan dalam menangani permasalahan prediksi time series. BPNN merupakan salah satu teknik dalam model jaringan syaraf tiruan yang bisa menangani permasalahan non linear tapi tidak bisa memberikan solusi yang unik dan global optima. Oleh karena itu BPNN sangat rentan mengalami permasalahan *overfitting* (Xie et al., 2006). Oleh karena itu dibutuhkan metode lain yang bisa menggantikan posisi BPNN dalam metode TEI@I.

*Adaptive Logic Network* (ALN) merupakan metode analitik baru yang diperkenalkan tahun 1996 yang telah dikembangkan untuk menangani berbagai macam permasalahan, termasuk *sales forecasting* dan *machine learning*. Sebagai salah satu tipe jaringan syaraf tiruan, ALN mempunyai keunggulan dibandingkan dengan pendekatan jaringan syaraf tiruan di masa lalu (Cogger dan Fanning, 1997). Pertama, ALN jauh lebih efisien dari sudut pandang komputasi. Kedua, hasilnya jauh lebih mudah untuk ditafsirkan. Ketiga, ALN mempunyai potensi untuk memeriksa

sifat jaringan untuk memenuhi spesifikasi yang dikehendaki yang tidak mungkin dilakukan dalam model jaringan syaraf tiruan sebelumnya.

*Neuro-fuzzy* mengambil keuntungan dari jaringan syaraf tiruan dalam proses pelatihan dan adaptasi dan keuntungan dari sistem logika fuzzy dalam menangani permasalahan *human-like reasoning* seperti penalaran pada tingkat lingusitik, transparansi dan *interpretability* dari model yang dihasilkan, serta penanganan data yang tidak tepat. *Neuro fuzzy* memungkinkan pembangunan sistem cerdas dalam tingkatan yang lebih tinggi (Palit dan Popovic, 2005).

Rojas dan Pomares (2004) melakukan perbandingan tiga metode *time series prediction* dalam melakukan *electricity load forecasting* yakni *Support Vector Machine* (SVM), *Adaptive Logic Networks* dan dua metode yang menggabungkan *parametric modelling* dan *neuro-fuzzy system*. Hasilnya, SVM memberikan nilai MAPE yang paling kecil yakni 1.98% disusul oleh ALN dengan nilai MAPE 2.14 % dan *neuro fuzzy system* menghasilkan nilai MAPE 2.49%.

SVM telah digunakan secara luas dalam berbagai bidang, mulai dari pengenalan pola hingga perhitungan regresi. Di samping itu, SVM merupakan salah satu algoritma yang cukup populer dalam dunia data mining. Berdasarkan IEEE ICDM (International Conference on Data Mining) yang dilaksanakan tahun 2006 di Hongkong, SVM masuk dalam 10 besar algoritma data mining yang paling dikenal (Wu dan Kumar, 2008).

## 2.2 Support Vector Machine

*Support Vector Machine* adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan bidang pemisah terbaik yang memisahkan dua buah *class input space*. Prinsip ini berkaitan erat dengan *statistical learning theory* dan merepresentasikan metodologi terbaru dalam pengembangan jaringan syaraf tiruan (Vapnik, 1998 dan 1995) untuk aplikasi perhitungan fungsi, perhitungan regresi dan *Signal Processing* (Vapnik et al., 1996). Termasuk juga pengenalan pola (Burgess, 1998), *Time Series Forecasting* (Cao, 2003), dan prediksi (Muller et al., 1997).

Pada awalnya, SVM didesain untuk menyelesaikan permasalahan pengenalan pola dengan menentukan suatu bidang pemisah terbaik yang berfungsi sebagai pemisah dua buah *class pada input space* dengan melakukan optimasi terhadap jarak di antaranya. Konsep ini didasarkan pada metode *Structural Risk Minimization* dan *statistical learning theory* dimana tingkat kesalahan pada data yang digunakan untuk pengujian dibatasi oleh tingkat kesalahan pada proses pelatihan dan dibatasi oleh *Vapnik-Chervonenkis dimension* (Palit dan Popovic, 2005).

SVM dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar SVM merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda dan Hart, 1973; Cover, 1965; Vapnik, 1964), kernel diperkenalkan oleh Aronszajn tahun 1950 dan demikian juga dengan konsep-konsep pendukung yang

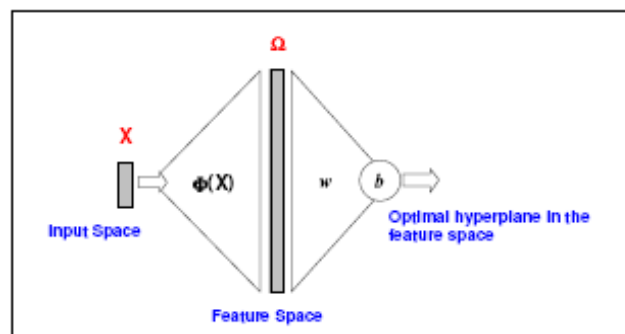


lain. Akan tetapi hingga tahun 1992 belum pernah ada upaya merangkaikan komponen-komponen tersebut.

SVM pada prinsipnya adalah *linear classifier*. Tetapi SVM bisa juga dipakai untuk permasalahan *non linear* dengan dua tahap. Pertama-tama data diproyeksikan ke ruang vektor baru berdimensi tinggi sedemikian hingga data itu dapat terpisah secara linear. Baru selanjutnya di "dunia baru" itu SVM mencari bidang pemisah optimal dengan cara yang sama sebagaimana di atas, yaitu bekerja sebagai *linear machine*. Ini didukung oleh teori Cober, yang menyatakan bahwa suatu ruang vektor dapat ditransformasikan ke ruang vektor baru yang pada probabilitas tinggi dapat terpisah secara linear, jika memenuhi dua syarat :

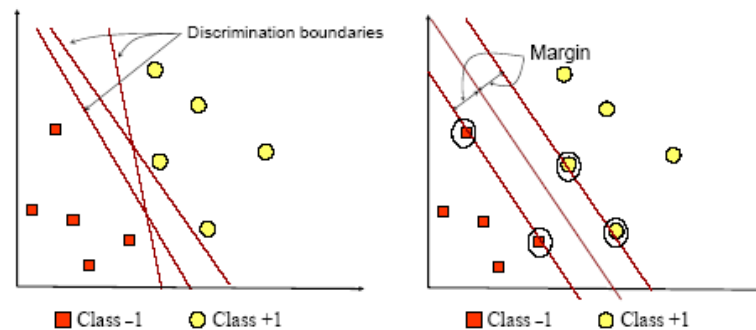
- (i) Transformasi itu non linear.
- (ii) Dimensi ruang vektor yang baru itu cukup tinggi.

Ilustrasi penggunaan konsep *Support Vector Machine* dapat dilihat pada gambar 2.1.



Gambar 2.1 Konsep *Support Vector Machine*

Berbeda dengan strategi jaringan syaraf tiruan yang berusaha mencari bidang pemisah antar kelas, SVM berusaha menemukan bidang pemisah yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier* dan selanjutnya dikembangkan agar dapat bekerja pada problem non linear dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Dewasa ini SVM telah berhasil diaplikasikan dalam problema dunia nyata (*real-world problems*) dan secara umum memberikan solusi yang lebih baik dibandingkan metode konvensional seperti misalnya *Artificial Neural Network*.



Gambar 2.2 SVM berusaha menemukan bidang pemisah terbaik yang memisahkan dua kelas -1 dan +1

Hampir sama seperti jaringan syaraf tiruan, SVM juga melakukan proses pelatihan dalam sistemnya. Pada SVM, tujuan dari proses pelatihan adalah untuk mencari posisi optimal dari bidang pemisah itu di ruang vektor. Parameter yang dipakai untuk mencari posisi optimal ini adalah "margin" yaitu jarak dari bidang pemisah ke dua buah kelas. Hasil perhitungan menunjukkan, bahwa bidang pemisah yang terbaik dicapai dengan memaksimalkan nilai margin. Posisi ini tercapai jika bidang pemisah itu terletak tepat di tengah-tengah, memisahkan kedua kelas. Ilustrasi

penemuan bidang pemisah terbaik ini dapat dilihat pada gambar 2.2. Pada SVM, fungsi klasifikasi dinyatakan dalam *dual form*, yaitu dinyatakan sebagai fungsi dari data pelatihan, bukan bobot. Proses pelatihan SVM sebagaimana dijelaskan di atas, bertujuan untuk mencari data mana dari set pelatihan yang paling informatif. Subset dari set pelatihan inilah hasil akhir dari pelatihan tersebut dan disebut dengan *support vector*. Di ruang vektor, *support vector* ini adalah data dari kedua buah kategori yang terletak paling dekat dengan bidang pemisah.

Dalam hal SVM, walaupun berbagai studi telah menunjukkan kelebihan metode SVM dibandingkan metode konvensional lain, SVM juga memiliki berbagai kelemahan (Nugroho et al., 2003). Kelebihan SVM antara lain sebagai berikut.

#### 1. Generalisasi

Generalisasi didefinisikan sebagai kemampuan suatu metode (SVM, jaringan syaraf tiruan dan sebagainya) untuk mengklasifikasikan suatu pola, yang tidak termasuk data yang dipakai dalam fase pelatihan metode itu. Vapnik menjelaskan bahwa error generalisasi dipengaruhi oleh dua faktor : error terhadap set pelatihan dan satu faktor lagi yang dipengaruhi oleh dimensi VC (Vapnik-Chervonenkis). Strategi pembelajaran pada jaringan syaraf tiruan dan umumnya metode *learning machine* difokuskan pada usaha untuk meminimalkan error pada set pelatihan. Strategi ini disebut *Empirical Risk Minimization* (ERM). Adapun SVM selain meminimalkan error pada set pelatihan, juga meminimalkan faktor kedua. Strategi ini disebut *Structural Risk Minimization* (SRM) dan dalam SVM diwujudkan dengan memilih bidang pemisah dengan margin terbesar.

Berbagai studi empiris menunjukkan bahwa pendekatan SRM pada SVM memberikan error generalisasi yang lebih kecil daripada yang diperoleh dari strategi ERM pada jaringan syaraf tiruan maupun metode yang lain. SRM bertujuan untuk menjamin batas atas dari generalisasi pada data pengujian dengan cara mengontrol "kapasitas" (fleksibilitas) dari hipotesis hasil pembelajaran. Untuk mengukur kapasitas ini digunakan dimensi Vapnik-Chervonenkis (VC).

## 2. *Curse of dimensionality*

*Curse of dimensionality* didefinisikan sebagai masalah yang dihadapi suatu metode pengenalan pola dalam mengestimasi parameter (misalnya jumlah *hidden neuron* pada jaringan syaraf tiruan, *stopping criteria* dalam proses pembelajaran dan sebagainya) dikarenakan jumlah sampel data yang relatif sedikit dibandingkan dimensional ruang vektor data tersebut. Semakin tinggi dimensi dari ruang vektor informasi yang diolah, membawa konsekuensi dibutuhkan jumlah data dalam proses pembelajaran. Pada kenyataannya seringkali terjadi data yang diolah berjumlah terbatas dan untuk mengumpulkan data yang lebih banyak tidak mungkin dilakukan karena kendala biaya dan kesulitan teknis. Dalam kondisi tersebut, jika metode itu "terpaksa" harus bekerja pada data yang berjumlah relatif sedikit dibandingkan dimensinya, akan membuat proses estimasi parameter metode menjadi sangat sulit. *Curse of dimensionality* sering dialami dalam aplikasi di bidang *biomedical engineering* karena biasanya data biologi yang tersedia sangat terbatas dan penyediaannya memerlukan biaya tinggi. Vapnik membuktikan bahwa tingkat generalisasi yang diperoleh oleh

SVM tidak dipengaruhi oleh dimensi dari input vektor (Vapnik, 1995). Hal ini merupakan alasan mengapa SVM merupakan salah satu metode yang tepat dipakai untuk memecahkan masalah berdimensi tinggi, dalam keterbatasan sampel data yang ada.

### 3. Landasan teori

Sebagai metode yang berbasis statistik, SVM memiliki landasan teori yang dapat dianalisa dengan jelas, dan tidak bersifat *black box*.

### 4. *Feasibility*

SVM dapat diimplementasikan relatif mudah, karena proses penentuan support vector dapat dirumuskan dalam QP problem. Dengan demikian jika kita memiliki *library* untuk menyelesaikan QP problem, dengan sendirinya SVM dapat diimplementasikan dengan mudah. Selain itu dapat diselesaikan dengan metode sekuensial sebagaimana penjelasan sebelumnya.

Lung (2006) menulis kelebihan SVM dalam berbagai penelitian dapat dilihat pada tabel 2.1.

Tabel 2.1 Kelebihan SVM dalam berbagai penelitian.

| Bidang  | Peneliti                                 | Kelebihan   |
|---|--|---|
| <i>Bioactivity prediction and compound classification</i> | Rahayu (2004)                            | <ul style="list-style-type: none"> <li>• Prediksi yang lebih baik pada data yang tidak tercatat.</li> <li>• Menawarkan solusi optimal yang unik untuk setiap permasalahan pelatihan.</li> <li>• Jumlah parameter lebih sedikit dibanding metode lain.</li> <li>• Mempunyai kinerja yang baik dalam pengolahan data yang mempunyai corak yang banyak berbeda.</li> </ul> |
| <i>Forecasting short-Term stock price indexes</i>         | Bao Rong Chang dan Shiou Fen Tsai (2002) | <ul style="list-style-type: none"> <li>• SVM menggunakan prinsip <i>Structural Risk Minimization</i>.</li> <li>• Membutuhkan kontrol parameter yang lebih kecil.</li> <li>• Menyediakan solusi global yang unik berdasarkan <i>quadratic programming</i>.</li> </ul>  |
| <i>Time series prediction via least square SVM</i>        | Zhu et al.. (2002)                       | <ul style="list-style-type: none"> <li>• Algoritma yang sederhana.</li> <li>• Konvergensi yang cepat.</li> <li>• Kemampuan yang baik dalam melakukan prediksi.</li> <li>• Mempunyai kinerja yang baik dalam data <i>time series</i> dengan <i>noise</i> yang tinggi.</li> </ul>   |
| <i>Air pollutant parameter forecasting using SVM</i>      | Lu et al.. (2002)                        | <ul style="list-style-type: none"> <li>• Mencapai level yang baik dalam kinerja generalisasi.</li> <li>• Perhitungan fungsi dengan meminimalisasi <i>upper bound</i> dan error generalisasi.</li> </ul>   |

Disamping kelebihanannya, SVM memiliki kelemahan atau keterbatasan, antara lain:

1. Sulit dipakai dalam problem berskala besar.

Skala besar dalam hal ini dimaksudkan dengan jumlah sample yang diolah.

2. SVM secara teoritik dikembangkan untuk problem klasifikasi dengan dua class.

Dewasa ini SVM telah dimodifikasi agar dapat menyelesaikan masalah dengan class lebih dari dua, antara lain strategi *One versus rest* dan strategi *Tree Structure*. Namun demikian, masing-masing strategi ini memiliki kelemahan, sehingga dapat dikatakan penelitian dan pengembangan SVM pada *multiclass-problem* masih merupakan tema penelitian yang masih terbuka.

### **2.2.1 Support Vector Machine untuk Permasalahan Klasifikasi**

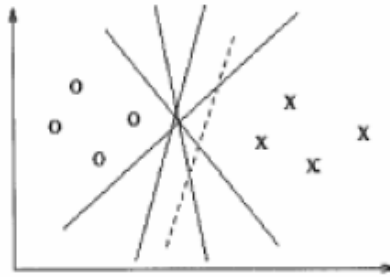
SVM adalah metode yang sangat *powerful* untuk klasifikasi data. Klasifikasi dilakukan dengan membuat bidang pemisah linear ataupun non linear pada ruang input data. Bidang pemisah ini hanya bergantung pada suatu subset dari data asli. Subset data ini, yang dibutuhkan untuk membangun bidang pemisah, menyusun *support vector*.

Pada permasalahan non linear, SVM berusaha untuk menempatkan pembatas linear di antara dua kelas berbeda dan mengaturnya agar jarak yang didapat bernilai maksimal (Vanajakshi dan Rilett, 2004).

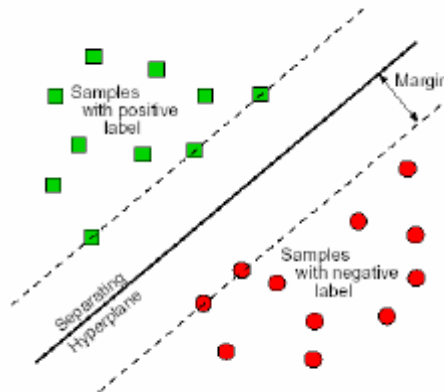
Untuk data yang terpisah secara linear, metode yang digunakan adalah menemukan bidang pemisah yang paling baik yang meminimalkan error saat

pelatihan. Selanjutnya batasnya diatur sedemikian rupa agar jarak antara batas dan data point terdekat bernilai maksimal.

Batas yang terpilih akan ditempatkan di tengah data, seperti pada gambar di bawah. Titik yang terletak paling dekat dengan bidang pemisah disebut dengan *support vector*. Hanya titik terdekat ini yang menentukan posisi bidang pemisah. Titik-titik yang lain tidak mempunyai pengaruh terhadap bidang pemisah. Jumlah bobot dari *support vector* adalah vektor normal *hyperplane*. Ilustrasi bidang pemisah dan *support vector* dapat dilihat pada gambar 2.3 dan 2.4.



Gambar 2.3 Bidang pemisah

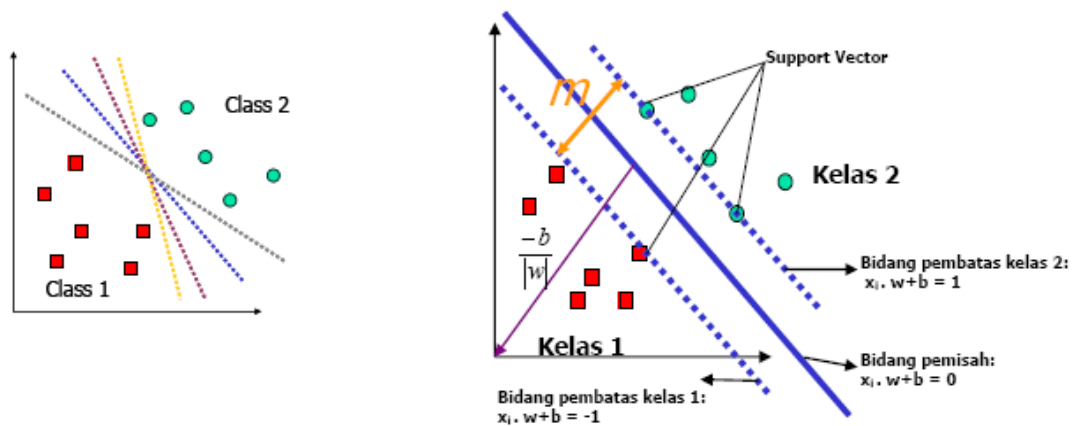


Gambar 2.4 *Support vector* dengan batas jarak maksimum



### 2.2.1.1 Perhitungan *Support Vector Machine* untuk Permasalahan Klasifikasi

Misalkan  $\{x_1, \dots, x_n\}$  adalah dataset dan  $y_i \in \{+1, -1\}$  adalah label kelas dari data  $x_i$ . Pada gambar 2.5 dapat dilihat berbagai alternatif bidang pemisah yang dapat memisahkan semua data set sesuai dengan kelasnya. Namun, bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar.



Gambar 2.5. Alternatif bidang pemisah (kiri) dan bidang pemisah terbaik dengan margin (m) terbesar (kanan).

Adapun data yang berada pada bidang pembatas ini disebut *support vector*. Dalam contoh di atas, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua sehingga diperoleh :

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \quad (2.1)$$

$$x_i \cdot w + b \leq -1 \text{ for } y_i = -1$$

$w$  adalah normal bidang dan  $b$  adalah posisi bidang relatif terhadap pusat koordinat. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak

garis ke titik pusat) adalah  $\frac{1-b-(1-b)}{w} = \frac{2}{|w|}$ . Nilai margin ini dimaksimalkan dengan tetap memenuhi (2.1). Dengan mengalikan  $b$  dan  $w$  dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstanta yang sama. Oleh karena itu konstrain (2.1) merupakan *scaling constraint* yang dapat dipenuhi dengan *rescaling*  $b$  dan  $w$ . Selain itu, karena memaksimalkan  $\frac{1}{|w|}$  sama dengan meminimumkan  $|w|^2$  dan jika kedua bidang pembatas pada (2.1) direpresentasikan dalam pertidaksamaan (2.2),

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad (2.2)$$

maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu

$$\min \frac{1}{2} |w|^2 \quad (2.3)$$

$$s. t \ y_i(x_i \cdot w + b) - 1 \geq 0$$

persoalan ini akan lebih mudah diselesaikan jika diubah ke dalam formula lagrangian menggunakan *lagrange multiplier*. Dengan demikian permasalahan optimasi konstrain dapat diubah menjadi

$$\min_{w,b} L_p(w, b, a) \equiv \frac{1}{2} |w|^2 - \sum_{i=1}^n a_i \cdot y_i(x_i \cdot w + b) + \sum_{i=1}^n a_i \quad (2.4)$$

dengan tambahan konstrain,  $a_i \geq 0$  (nilai dari koefisien *lagrange*). Dengan meminimumkan  $L_p$  terhadap  $w$  dan  $d$ , maka dari  $\frac{\partial}{\partial b} L_p(w, b, a) = 0$  diperoleh (2.5) dan dari  $\frac{\partial}{\partial w} L_p(w, b, a) = 0$  diperoleh (2.6).

$$\sum_{i=1}^n a_i \cdot y_i = 0 \quad (2.5)$$

$$w = \sum_{i=1}^n a_i \cdot y_i \cdot x_i \quad (2.6)$$

Vektor  $w$  seringkali bernilai besar (mungkin tak terhingga), tetapi nilai  $a_i$  terhingga. Untuk itu, formula *lagrangian*  $L_p$  (*primal problem*) diubah ke dalam *dual problem*  $L_D$ . Dengan mensubstitusikan persamaan (2.6) ke  $L_p$  diperoleh *dual problem*  $L_D$  dengan konstrain berbeda.

$$L_D(a) \equiv \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j x_i \cdot x_j \quad (2.7)$$

$\min_{w, b} L_P = \max_a L_D$ . Jadi persoalan pencarian bidang pemisah terbaik dapat dirumuskan sebagai berikut :

$$\max_a L_D \equiv \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j x_i \cdot x_j \quad (2.8)$$

$$s.t \sum_{i=1}^n a_i y_i = 0, a_i \geq 0$$

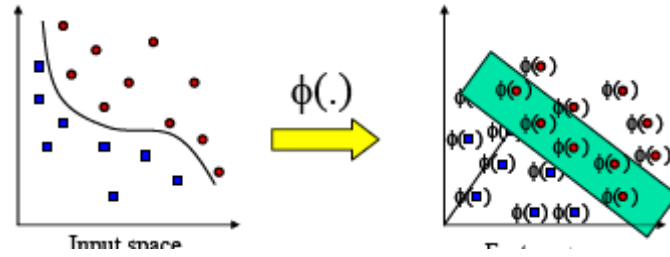
Dengan demikian, dapat diperoleh  $a_i$  yang nantinya digunakan untuk menemukan  $w$ . Terdapat nilai  $a_i$  untuk setiap data pelatihan. Data pelatihan yang memiliki nilai  $a_i > 0$  adalah *support vector* sedangkan sisanya memiliki nilai  $a_i = 0$ . Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Formula pencarian bidang pemisah terbaik ini adalah permasalahan *quadratic programming*, sehingga nilai maksimum global dari  $a_i$  selalu dapat ditemukan. Setelah solusi permasalahan *quadratic programming* ditemukan (nilai  $a_i$ ), maka kelas dari data pengujian  $x$  dapat ditentukan berdasarkan nilai dari fungsi keputusan :

$$f(x_d) = \sum_{i=1}^{ns} a_i y_i x_i \cdot x_d + b \quad (2.9)$$

$x_i$  adalah *support vector*,  $ns$  = jumlah *support vector* dan  $x_d$  adalah data yang akan diklasifikasikan.

Persamaan (2.9) hanya bisa digunakan untuk menyelesaikan permasalahan klasifikasi linear dalam SVM. Pada permasalahan non linear, data yang ada harus ditransformasikan ke dalam dimensi ruang fitur (*feature space*) sehingga dapat dipisahkan secara linear pada *feature space*. Ilustrasi pemetaan input ke *feature space* dapat dilihat pada gambar 2.6.



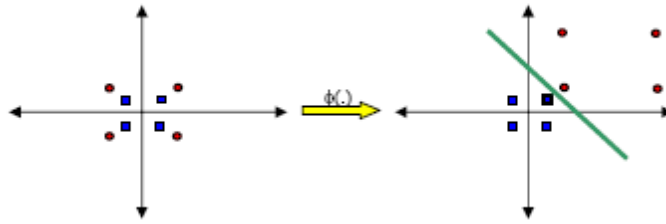
Gambar 2.6 Input SVM dan *feature space*

Caranya, data dipetakan dengan menggunakan fungsi pemetaan (transformasi)  $x_k \rightarrow \phi(x_k)$  ke dalam *feature space* sehingga terdapat bidang pemisah yang dapat memisahkan data sesuai dengan kelasnya.

Misalkan terdapat data set yang datanya memiliki dua atribut dan dua kelas yaitu kelas positif dan negatif. Data yang memiliki kelas positif adalah  $\{(2,2),(2,-2),(-2,2),(-2,-2)\}$ , dan data yang memiliki kelas negative  $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$ . Apabila data ini digambarkan dalam ruang dua dimensi (gambar 2.7) dapat dilihat data ini tidak dapat dipisahkan secara linear. Oleh karena itu, digunakan fungsi transformasi berikut:

$$\phi(x_1, x_2) = \begin{cases} \sqrt{x_1^2 + x_2^2} > 2 \rightarrow (4 - x_2 + |x_1 - x_2|, 4 - x_1 + |x_1 - x_2|) \\ \sqrt{x_1^2 + x_2^2} \leq 2 \rightarrow (x_1, x_2) \end{cases} \quad (2.10)$$

Data sesudah transformasi adalah  $\{(2,2),(6,2),(6,6),(2,6)\}$  untuk kelas negative dan  $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$  untuk kelas positif. Selanjutnya pencarian bidang pemisah terbaik dilakukan pada data ini.



Gambar 2.7. Contoh transformasi untuk data yang tidak dapat dipisahkan secara linear

Dengan menggunakan fungsi transformasi  $x_k \rightarrow \Phi(x_k)$  maka nilai  $w = \sum_{i=1}^n a_i \cdot y_i \Phi(x_i)$  dan fungsi hasil pelatihan yang dihasilkan adalah

$$f(x_d) = \sum_{i=1}^{ns} a_i y_i \Phi(x_i) \Phi(x_d) + b \quad (2.11)$$

*Feature space* dalam prakteknya biasanya memiliki dimensi yang lebih tinggi dari vektor input (*input space*). Hal ini mengakibatkan komputasi pada *feature space* mungkin sangat besar, karena ada kemungkinan *feature space* dapat memiliki jumlah *feature* yang tidak terhingga. Selain itu, sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan "*kernel trick*". Dari persamaan (2.11) dapat dilihat terdapat *dot product*  $\Phi(x_i) \Phi(x_d)$ . Jika terdapat sebuah fungsi kernel  $K$  sehingga  $K(x_i, x_d) = \Phi(x_i) \Phi(x_d)$ , maka fungsi transformasi  $\Phi(x_k)$  tidak perlu diketahui secara persis. dengan demikian fungsi yang dihasilkan dari pelatihan adalah

$$f(x_d) = \sum_{i=1}^{ns} a_i y_i K(x_i, x_d) + b \quad (x_i = \text{support vector}) \quad (2.12)$$

Syarat sebuah fungsi untuk menjadi fungsi kernel adalah memenuhi *teorema Mercer* yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat *positive semi-definite*.

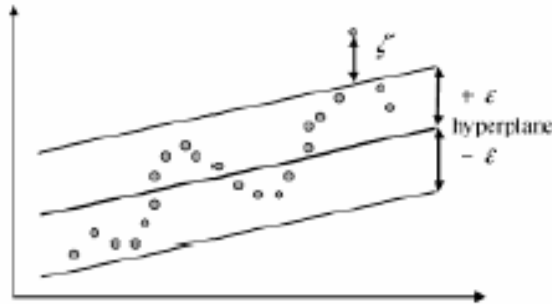
### 2.2.2 Support Vector Machine untuk Permasalahan Regresi

Aplikasi SVM dalam *time series forecasting* dikenal dengan istilah *Support Vector Regression* (SVR). SVR merupakan salah satu pendekatan dalam menyelesaikan permasalahan regresi (Smola et al., 1998). SVR merupakan varian dari SVM yang dikembangkan oleh Vapnik. Saat ini, SVR telah berhasil diaplikasikan dalam *time series prediction* (Mukherjee et al., 1997) dan *financial forecasting* (Tay dan Cao, 2001).

Ide dasar permasalahan regresi adalah menentukan fungsi yang bisa memperkirakan nilai di masa mendatang secara akurat. Dalam regresi, tujuan utamanya adalah membentuk bidang pemisah yang terletak dekat dengan semakin banyak data point yang memungkinkan. Oleh karena itu, tujuan utama dari SVM adalah menentukan bidang pemisah dengan ukuran terkecil sekaligus meminimalkan jumlah dari jarak antara data poin ke bidang pemisah (Theodore et al., 2000).

SVR berbeda dengan penggunaan SVM dalam permasalahan klasifikasi dengan memperkenalkan fungsi alternatif loss yang telah dimodifikasi untuk

mengukur jarak. Terlebih lagi, parameter yang mempengaruhi kualitas regresi adalah nilai error  $C$ , lebar tube  $\epsilon$  dan fungsi pemetaan. Ilustrasi parameter SVR dapat dilihat pada gambar 2.8.



Gambar 2.8 SVR bekerja dengan variabel  $C$  dan sembarang fungsi  $\epsilon$ . Variabel  $C$  digunakan untuk menghitung nilai error di luar jangkauan  $\epsilon$ .

Ide dasar dari SVR adalah memetakan data ke dalam ruang vektor berdimensi tinggi melalui pemetaan non linear dan melakukan regresi linear dalam ruang tersebut. Metode *support vector* juga bisa diaplikasikan langsung ke dalam regresi, dengan menjaga karakteristik algoritma margin maksimal : fungsi non linear dipelajari dengan menggunakan fungsi kernel yang membentuk ruang vektor berdimensi tinggi dimana sistem dikontrol melalui parameter yang tidak tergantung pada dimensi ruang vektor (Cristianini dan Shawetaylor, 2000).

#### 2.2.2.1 Perhitungan *Support Vector Machine* untuk Permasalahan Regresi

SVM untuk permasalahan regresi mirip dengan penggunaan SVM pada permasalahan non linear. Hanya saja perlu adanya penambahan variabel pada pemecahan permasalahan kali ini. Kedua bidang pembatas (2.1) harus diubah



sehingga lebih fleksibel (untuk kondisi tertentu) dengan penambahan variable

$\xi_i$  ( $\xi_i \geq 0, \forall_i : \xi_i = 0$  jika  $x_i$  diklasifikasikan dengan benar) menjadi

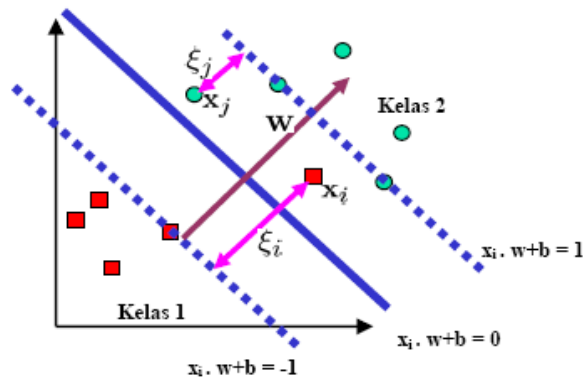
$x_i \cdot w + b \geq 1 - \xi_i$  untuk kelas 1 dan  $x_i \cdot w + b \leq -1 + \xi_i$  untuk kelas 2.

Pencarian bidang pemisah terbaik dengan penambahan variabel  $\xi_i$  sering juga disebut dengan *soft margin hyperplane*. Dengan demikian formula pencarian bidang pemisah terbaik menjadi :

$$\min \frac{1}{2} |w|^2 + C \left( \sum_{i=1}^n \xi_i \right) \quad (2.13)$$

$$s. t \ y_i(x_i \cdot w + b) \geq 1 - \xi_i$$

C adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Bentuk persoalan (2.13) memenuhi prinsip SRM, dimana meminimumkan  $\frac{1}{2} |w|^2$  ekuivalen dengan meminimumkan dimensi VC dan meminimumkan  $C(\sum_{i=1}^n \xi_i)$  berarti meminimumkan error pada data pelatihan.



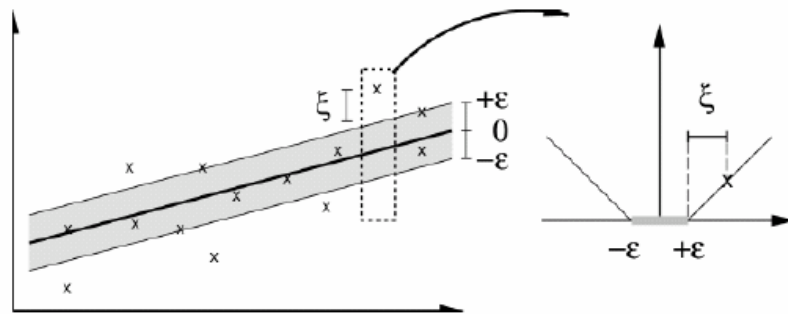
Gambar 2.9. *Soft margin hyperplane*

Fungsi perhitungan SVR ditunjukkan sebagai berikut

$$f(x) = (w \cdot (x)) + b \quad (2.14)$$

$\varepsilon$  -insensitive loss function dibentuk melalui fungsi

$$|y_i - (w \cdot x_i + b)|_\varepsilon = \begin{cases} 0 & \text{if } |y_i - (w \cdot x_i + b)| \leq \varepsilon \\ |y_i - (w \cdot x_i + b)| - \varepsilon & \text{otherwise} \end{cases} \quad (2.15)$$



Gambar 2.10.  $\varepsilon$  insensitive loss

Dengan mensubstitusikan persamaan 2.13 dan 2.15

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m |y_i - f(x_i)|_\varepsilon \quad (2.16)$$

permasalahan ini bisa ditransformasikan ke dalam permasalahan optimasi konstrain dengan menggunakan *slack variable*. Dibutuhkan dua tipe *slack variable*  $f(x_i) - y_i > \varepsilon$  dan  $y_i - f(x_i) > \varepsilon$ , didefinisikan dengan  $\xi$  dan  $\xi^*$ . Permasalahan optimasinya

$$\min_{w \in \mathcal{H}, \xi^{(*)} \in \mathbb{R}, b \in \mathbb{R}} \tau(w, \xi^{(*)}) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*) \quad (2.17)$$

$$\text{subject to } f(x_i) - y_i \leq \varepsilon + \xi_i \quad (2.18)$$

$$y_i - f(x_i) \leq \varepsilon + \xi_i^* \quad (2.19)$$

$$\xi_i, \xi_i^* \geq 0 \text{ for all } i = 1, \dots, m \quad (2.20)$$

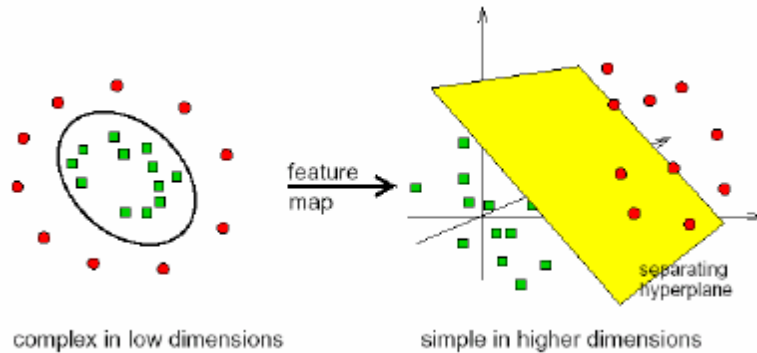
Dengan menggunakan *lagrange multipliers*, permasalahan optimasi konstrain dapat diubah menjadi

$$\begin{aligned} \max_{a, a^* \in \mathbb{R}^m} W(a, a^*) = & -\varepsilon \sum_{i=1}^m (a_i^* + a_i) + \sum_{i=1}^m (a_i^* - a_i) y_i \\ & - \frac{1}{2} \sum_{i,j=1}^m (a_i^* - a_i)(a_j^* - a_j) k(x_i, x_j) \end{aligned} \quad (2.21)$$

$$\text{subject to } 0 \leq a_i, a_i^* \leq C \text{ for all } i = 1, \dots, m \text{ and } \sum_{i=1}^m (a_i^* - a_i) = 0 \quad (2.22)$$

Lagrange multipliers  $\alpha_i^*$ ,  $\alpha_i$  merepresentasikan solusi untuk permasalahan quadratic. Hanya nilai tidak nol dari Lagrange multipliers pada (2.19) yang berguna dalam peramalan garis regresi dan dikenal dengan istilah *support vectors* (Wu et al., 2004). Nilai konstan C menentukan penalti perhitungan error.

### 2.2.3 Fungsi Kernel



Gambar 2.11 Fungsi Pemetaan

Pada permasalahan klasifikasi, lebih mudah dilakukan dalam ruang vektor berdimensi tinggi dimana akan dibangun bidang pemisah dengan margin terbesar. Konstruksi bidang pemisah tergantung dari *inner products* yang digunakan dalam membangun ruang vektor. Oleh karena itu digunakan fungsi kernel pada dimensi rendah yang bisa merepresentasikan permasalahan dalam ruang vektor berdimensi tinggi. Wujud transformasi ke ruang dimensi yang lebih tinggi tidak perlu diketahui secara detail. Bagaimana cara pemetaan satu per satu suatu titik sehingga berada pada suatu titik pada dimensi yang lebih tinggi tidak perlu diketahui. Umumnya data akan diubah ke dimensi ruang yang jauh lebih tinggi daripada dimensi aslinya (beberapa puluh bahkan bisa beberapa ratus). Dan semua itu cukup dilakukan dengan menggunakan fungsi kernel. Yang penting disini adalah data bisa dipisahkan secara linear pada ruang yang lebih tinggi.

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang ada yang bersifat *linear separable*. Kebanyakan bersifat non linear. Untuk

menyelesaikan problem non linear, SVM dimodifikasi dengan memasukkan fungsi kernel (Nugroho et al., 2003).

Konsep kernel ini bukan hal yang baru, dan dibahas oleh Aizerman di tahun 1964. Tetapi tidak banyak yang memperhatikan konsep ini, sampai Boser, Guyon dan Vapnik memakainya dalam paper SVM mereka tahun 1992. Konsep SVM sebenarnya adalah kombinasi harmonis dari teori-teori di statistik jaman dulu.

Konsep kernel memang lebih luas daripada *Support Vector Machine*. Fungsi kernel adalah satu cara merepresentasikan data, ke ruang vektor yang lebih tinggi, sehingga dapat meningkatkan *computational power* dari suatu *linear machine*. Umumnya representasi dari suatu data merupakan fungsi yang secara langsung diterapkan pada data itu, misalnya pemetaan  $x$  dapat dirumuskan sebagai  $F(x)$ . Tapi kernel memperkenalkan pendekatan lain, dimana data  $x$  itu tidak direpresentasikan secara individual, melainkan dalam bentuk *pairwise comparison*, yang menunjukkan "similarity" antara satu data dengan yang lain. Misalnya ada 10 data, maka akan diperoleh kernel matrix berukuran  $10 \times 10$  yang merepresentasikan seberapa jauh similarity antara satu data dengan data yang lain. Secara sederhana metode kernel dapat didefinisikan sebagai metode yang algoritmanya hanya mengolah *similarity matrix* yang didefinisikan oleh sebuah kernel.

Dalam SVM, *decision function*  $f(x)$  dinyatakan dalam bentuk kombinasi linear antara vektor  $x$ , dengan satu subset dari set pelatihan yang disebut *support vector*. *Support vector* ini merupakan subset data yang paling "informatif", yaitu yang terletak paling dekat dengan *classification border*. Dalam penentuan *support vector*,

komputasi dapat dinyatakan dalam bentuk kombinasi satu pola dengan yang lain, dalam bentuk *dot product*. Ini adalah kernel yang paling dasar, disebut sebagai linear kernel. Dalam bentuk lebih umum, dapat dipakai berbagai fungsi sebagai kernel, seperti Gaussian, Polynomial, dan sebagainya.

Dalam aplikasinya ke non linear problem, pertama-tama data diproyeksikan ke ruang vektor yang berdimensi lebih tinggi. Selanjutnya *linear discrimination* dilakukan pada ruang vektor baru tersebut, yang mana bidang pemisah itu jika dikembalikan ke ruang vektor awal, akan menghasilkan *non-linear classification border*. Komputasi dot product di ruang vektor baru tersebut dilakukan secara implisit, yaitu dengan fungsi kernel pada data awal.

Dengan penjelasan secara formulasi, pemilihan fungsi pada SVM memerlukan perhitungan dot product dari datanya :  $(x_i, x_j)$ . Saat diproyeksikan ke ruang vektor berdimensi tinggi, perhitungan tersebut tentunya akan berubah menjadi  $(H(x_i), H(x_j))$  dimana  $H$  adalah fungsi yang mentransformasikan data awal  $x_i$  ke ruang vektor berdimensi tinggi. Perhitungan seperti ini biasanya dilakukan dengan menghitung terlebih dahulu nilai  $H(x_i)$  dan  $H(x_j)$ , baru kemudian dilakukan perhitungan dot product-nya. Dengan kata lain fungsi  $H$  harus didefinisikan terlebih dahulu. Dalam SVM hal ini tidak diperlukan. Fungsi  $H$  tidak perlu diketahui. Perhitungan  $H(x_i), H(x_j)$  tetap memakai  $x_i$  dan  $x_j$  yang nantinya dimasukkan ke dalam fungsi yang disebut fungsi kernel. Hal itu yang disebut dengan "kernel trick", menghitung dot product di "dunia baru" secara implisit. Proses pengaturan parameter di jaringan syaraf tiruan (MLP-BP) dilakukan dengan memvariasi banyaknya *hidden*

*neuron*, nilai *learning rate*, momentum, *stopping criterion* dan sebagainya. Di SVM, proses pengaturan parameter digantikan menjadi upaya memilih fungsi kernel yang optimal (Nugroho, 2006).

Berbagai macam studi mengenai pemilihan fungsi kernel yang disebabkan data yang tidak linear telah banyak dilakukan, diantaranya *Load Forecasting* (Chen et al., 2004), *Traffics Speed Prediction* (Vanajakshi dan Rilett, 2004) , *Travel Time Prediction* (Wu et al., 2004). Ada beberapa tipe fungsi kernel yang digunakan dalam SVM. Yang biasa digunakan adalah Linear, Polynomial, Radial Basis Function dan Sigmoid Kernel Function. Persamaan untuk masing-masing fungsi kernel dapat dilihat pada tabel 2.2.

Tabel 2.2 Fungsi Kernel

| Fungsi Kernel | Persamaan  |
|---------------|--|
| Linear        | $K(x,y) = x \cdot y$                                     |
| Polynomial    | $K(x,y) = (x \cdot y + 1)^p$                             |
| RBF           | $K(x,y) = \exp\left(-\frac{\ x-y\ ^2}{2\sigma^2}\right)$ |
| Sigmoid       | $K(x,y) = \tanh(kx \cdot y - \delta)$                    |

Akan tetapi tidak ada jaminan jika satu fungsi kernel akan menghasilkan kinerja yang baik untuk semua dataset. Pemilihan fungsi kernel sangat bergantung pada permasalahan dan parameter yang ada.

Karatzoglou et al. (2006) menyatakan bahwa kernel gaussian, Laplace RBF dan Bessel merupakan kernel yang biasa digunakan untuk keperluan umum ketika tidak ada pengetahuan tentang data. Kernel Linear biasa digunakan jika data vektor yang digunakan berukuran besar seperti yang biasanya terjadi dalam kategorisasi teks. Kernel Polynomial populer dalam *image processing* dan kernel Sigmoid digunakan sebagai proxy dalam jaringan syaraf tiruan. Kernel Splines dan ANOVA RBF biasanya mempunyai kinerja yang baik dalam permasalahan regresi.

Lin dan Lin (2004) pernah melakukan penelitian mengenai penggunaan sigmoid kernel dalam SVM. Dalam eksperimennya dengan beberapa tipe data set yang berbeda, kinerja fungsi sigmoid berada pada tingkat yang sama seperti fungsi kernel lain seperti RBF. Akan tetapi pemilihan parameter yang tepat untuk fungsi sigmoid sangat sulit untuk dilakukan, oleh karena itu Lin dan Lin (2004) tidak menyarankan penggunaan fungsi sigmoid.

Pada penelitian yang dilakukan Lin dan Lin (2004), terlihat bahwa parameter pada kernel sigmoid mempunyai kemiripan seperti pada kernel RBF dalam hal kinerja. Penelitian menunjukkan bahwa kinerja keduanya hampir sama. Di antara beberapa fungsi kernel yang ada, disarankan RBF digunakan sebagai pilihan utama untuk pengguna secara umum (Lin dan Lin, 2004).

Sebagai perbandingan, Frontzek et al. (2001) telah membuat perbandingan kinerja di antara tiga kernel berbeda, yaitu RBF, Sigmoid, dan Polynomial. Kinerja kernel juga dibandingkan dengan jaringan MLP dan RBF lain dalam permasalahan prediksi non linear jaringan biologi.



Berdasarkan hasil yang didapatkan, RBF kernel menunjukkan hasil yang memuaskan dibandingkan fungsi kernel lain dan model jaringan syaraf tiruan dengan nilai error 0.199 untuk RBF dan 0.296 untuk kernel polynomial. Akan tetapi dalam kasus waktu komputasi, kernel polynomial mempunyai kinerja yang lebih baik dibandingkan kernel RBF. Oleh karena itu, Frontzek et al. (2001) menyimpulkan bahwa kernel RBF dan kernel polinomial bisa digunakan untuk mempelajari data biologis non linear sedangkan fungsi sigmoid tidak bisa melakukannya.

## 2.3 Pemodelan Data Time Series

*Data preprocessing* adalah proses transformasi data asli untuk mendapatkan beberapa informasi dalam data time series seperti tren dan musim serta untuk menghilangkan karakteristik non stasioner (Athanasios, 2005). Berdasarkan Chen et al. (2004) ada beberapa hal yang harus diperhatikan dalam membentuk model SVM untuk melakukan prediksi *time series*, yaitu:

1. Pemilihan *Feature*
2. Segmentasi Data
3. Representasi Data

### 2.3.1 Pemilihan Feature

Dalam training dan testing set, ada komponen yang disebut dengan *feature* atau atribut ( $x_i$ ) yang akan digunakan untuk melakukan prediksi terhadap target ( $y_t$ ). Pemilihan *feature* untuk prediksi didasarkan pada data sebelumnya. Jadi untuk

prediksi data *time series*, *feature* atau atribut ( $x_i$ ) berisi data sebelumnya yang telah dikumpulkan, bisa 5, 10, 15 atau 20 hari. Dalam Lung (2006), Thomason menjelaskan bahwa dalam kasus ini jumlah data *time series* yang diperlukan bukanlah suatu hal yang sangat penting, tapi akan lebih baik jika lebih panjang dari lima hari. Chen et al. (2004) menggunakan tujuh data *time series* untuk melakukan prediksi dengan pertimbangan dalam tujuh hari itu distribusi datanya cukup merata.

Memilih *forecasting horizon* yang tepat merupakan langkah pertama dalam melakukan peramalan keuangan (Cao, 2003). Dari aspek perdagangan, *forecasting horizon* harus cukup panjang sehingga nilai yang berlebihan dapat dihindarkan. Dari aspek prediksi, *forecasting horizon* harus cukup pendek untuk menjaga persistensi *financial time series* dalam durasi yang terbatas.

Pada pelatihan dengan menggunakan SVM tidak terdapat mekanisme untuk mengetahui atribut yang penting atau yang kurang penting. Atribut yang kurang penting umumnya tidak mempengaruhi efektifitas SVM. Oleh karena itu, jika atribut yang kurang penting ini dibuang maka efisiensi SVM akan meningkat (Sembiring, 2007).

### 2.3.2 Segmentasi Data

Setelah dilakukan pemilihan *feature*, akan lebih baik jika data *time series* yang ada disegmentasi sesuai dengan karakteristiknya karena data *time series* yang ada bisa menunjukkan pola *time series*. Chen et al. (2004) menunjukkan bahwa segmentasi data bisa meningkatkan performa model karena data *time series*

mempunyai distribusi yang berbeda pada waktu yang berbeda pula. Menggunakan semua data *time series* yang ada tanpa segmentasi bisa memberikan lebih banyak informasi tapi juga beresiko memberikan *noise* yang lebih besar sehingga mempengaruhi hasil prediksi. Chen et al. (2004) menunjukkan pada penelitiannya tentang *electricity load forecasting* bahwa model yang dibangun dengan data pelatihan pada *winter* bisa memberikan hasil prediksi yang baik daripada menggunakan data pelatihan pada satu tahun penuh, tentunya dengan catatan bahwa data pada *winter* itu harus cukup banyak sehingga mampu memberikan informasi yang cukup untuk membangun model yang baik.

### 2.3.3 Representasi Data

Selanjutnya setelah memilih *feature* yang sesuai dan melakukan segmentasi data yang tepat terhadap data time series fokus berikutnya adalah menghasilkan representasi data yang tepat sesuai dengan format data yang dibutuhkan oleh sistem untuk membentuk model SVM.

Di samping itu, tindakan penting yang harus dilakukan dalam mempersiapkan set data untuk model SVM adalah normalisasi data. Normalisasi data adalah proses persiapan data untuk digunakan langsung dalam sistem. Normalisasi digunakan agar data berada pada *range* yang disyaratkan oleh sistem. Untuk data yang *range*-nya besar jelas perlu dilakukan normalisasi. Adapun dengan data yang *range*-nya kecil perlu juga dilakukan normalisasi agar efeknya dalam proses pelatihan sama dengan

*feature* yang lain. Normalisasi yang dilakukan pada training data harus juga dilakukan dengan cara yang sama pada testing data (Nugroho, 2008). Misalnya normalisasi dilakukan dengan mengurangi suatu nilai *feature* (misalnya  $x$ ) dengan nilai mean *feature* dan kemudian dibagi dengan *standard deviation*-nya dimana mean dan *standard deviation*-nya dihitung dari training data. Maka selanjutnya pada testing data juga dilakukan normalisasi yang sama dengan nilai mean dan *standard deviation* yang sama seperti pada normalisasi training data. Hal ini sangat penting karena akan menjamin bahwa training data dan testing data ditransformasikan linear oleh operasi yang sama.

Hsu et al. (2003) mendeskripsikan pentingnya *data scaling* untuk menghilangkan pengaruh *feature* yang memiliki *range* besar mendominasi *feature* yang memiliki *range* yang kecil. Hsu et al. (2003) menyarankan skala range data antara  $[-1, +1]$  atau  $[1, 0]$ . *Data scaling* juga memberikan keuntungan dalam melakukan operasi perhitungan (Lung, 2006).

Sarle (2002) menjelaskan normalisasi dan *data scaling* sebenarnya mempunyai definisi yang berbeda. Normalisasi berarti membagi vektor dengan norma vektor (misalnya membuat semua unsur terletak antara 0 dan 1). *Data scaling* berarti menambah, mengurangi, mengalikan atau membagi vektor dengan nilai konstan seperti mengganti pengukuran suatu data (misal mengubah suhu dari Celcius ke Fahrenheit). Hsuan (2009) menyatakan bahwa *data scaling* tidak dibutuhkan tapi direkomendasikan sehingga tidak ada keharusan untuk melakukan *data scaling* apabila data sudah berada pada *range* yang tidak begitu besar.

## 2.4 Pemilihan Parameter

Parameter yang dipilih pada proses pelatihan untuk membentuk model SVM yang digunakan dalam proses prediksi dilakukan dengan metode trial and error. Pemilihan parameter akan mempengaruhi kinerja SVM model dalam melakukan prediksi. Beberapa parameter penting yang berkaitan dengan *Support Vector Machine* antara lain nilai  $C$ , epsilon dan fungsi kernel.

Nilai  $C$  menentukan hukuman untuk kesalahan estimasi (Wu et al., 2004). Semakin besar nilai  $C$  akan memberikan hukuman yang lebih tinggi untuk kesalahan estimasi yang pada akhirnya akan meminimalkan error dengan generalisasi yang lebih rendah dan begitu pula untuk nilai  $C$  yang lebih kecil. Jika nilai  $C$  mendekati tak hingga, model SVM yang kompleks akan dihasilkan sedemikian rupa sehingga SVM tidak mengijinkan terjadinya kesalahan.

Berdasarkan Wu et al. (2004), radius  $\epsilon$  menentukan data di dalam tube yang diabaikan dalam regresi. SVM untuk regresi berkaitan dengan slack variable  $C$  dan  $\epsilon$  *insensitive loss function*. Slack variable  $C$  digunakan untuk mengukur kesalahan di luar tabung  $\epsilon$ . Dalam Lung (2006), Cherkassky dan Mulier merekomendasikan bahwa nilai  $\epsilon$  harus lebih kecil untuk ukuran sampel yang lebih besar. Studi yang dilakukan Lu et al. (2002) menemukan bahwa nilai  $\epsilon$  mempunyai pengaruh yang kecil terhadap variasi kesalahan prediksi dan jumlah *support vector*. Selain itu, mereka menyarankan nilai  $\epsilon$  harus ditetapkan pada nilai yang kecil.

Semakin besar nilai epsilon, semakin sedikit jumlah *support vector* yang dipilih. Dengan kata lain, semakin besar nilai epsilon, akan menghasilkan estimasi yang cenderung 'flat' (Anonim, 2003).

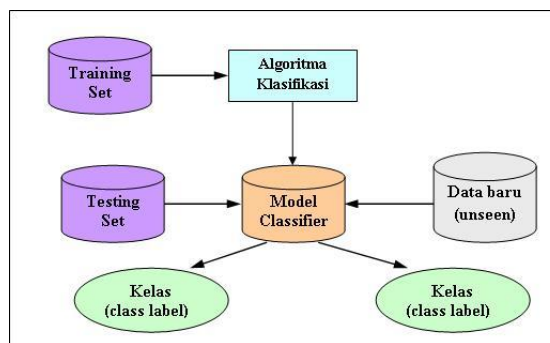
Pemilihan fungsi kernel merupakan hal yang sangat penting karena memungkinkan perkalian *dot product* pada ruang fitur dimensi tinggi tanpa harus mengetahui transformasi nonlinear. Dalam penentuan fungsi kernel ini sendiri ada beberapa parameter yang harus diperhatikan nilainya agar bisa menghasilkan arsitektur model SVM yang optimal dalam melakukan prediksi, yakni degree dan gamma.

Untuk memilih nilai degree dilakukan dengan metode trial dan error. Namun berdasarkan studi yang dilakukan oleh Ali dan Smith (2003) dengan menggunakan berbagai macam sample set dengan atribut dan ukuran yang berbeda, ruang pencarian nilai degree pada Polynomial kernel harus berkisar antara 2 dan 5. Sedangkan berdasarkan Lung (2006) melakukan penelitian dengan menggunakan nilai degree antara 0 dan 5.

Pemilihan nilai gamma bervariasi di antara beberapa peneliti time series prediction, yang berkisar dari 0.0625 hingga 100. Namun, Lu et al. (2002) menyatakan bahwa nilai  $\gamma$  mempengaruhi tingkat akurasi prediksi. Pernyataan ini didasarkan pada penelitian mereka yang menemukan bahwa memilih  $\gamma \rightarrow 0$  atau  $\gamma \rightarrow \infty$  akan memberikan hasil generalisasi yang buruk.

## 2.5 Set Pelatihan, Set Pengujian dan Validasi

Ketiga istilah ini sangat berkaitan dengan klasifikasi data. Sebuah model *classifier* pada klasifikasi data dibentuk berdasarkan data yang sudah ada, dan kemudian model tersebut digunakan untuk klasifikasi dan prediksi data baru yang belum pernah ada. Ilustrasinya dapat dilihat pada gambar 2.12.



Gambar 2.12 Proses klasifikasi

Data umumnya dibagi menjadi set pelatihan dan set pengujian. Set pelatihan digunakan oleh algoritma klasifikasi (misalnya: *decision tree*, bayesian, jaringan syaraf tiruan, SVM) untuk membentuk sebuah model *classifier*. Model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada. Set pengujian digunakan untuk mengukur sejauh mana *classifier* berhasil melakukan klasifikasi dengan benar. Karena itu, data yang ada pada set pengujian seharusnya tidak boleh ada pada set pelatihan sehingga dapat diketahui apakah model *classifier* sudah “pintar” dalam melakukan klasifikasi.

Lain lagi halnya dengan set validasi. Umumnya beberapa algoritma klasifikasi memerlukan beberapa parameter. Misalnya: jumlah *hidden layer* dan

*learning rate* pada jaringan syaraf tiruan; parameter kernel pada SVM. Biasanya sebagian dari set pelatihan diambil untuk set validasi. Set validasi ini digunakan untuk mencari parameter yang paling baik untuk sebuah algoritma klasifikasi (Prasetyo, 2006).

Memisahkan data menjadi set pelatihan dan set pengujian dimaksudkan agar model yang diperoleh nantinya memiliki kemampuan generalisasi yang baik dalam melakukan klasifikasi data. Tidak jarang sebuah model klasifikasi dapat melakukan klasifikasi data dengan sangat baik pada training set, tetapi sangat buruk dalam melakukan klasifikasi data yang baru dan belum pernah ada. Hal ini dinamakan *overfitting*.

Afiahayati (2008) menjelaskan *overfitting* sebagai keadaan dimana model hasil pelatihan sangat baik mendeskripsikan set pelatihan, namun kurang bisa menangkap pola-pola yang muncul diluar set pelatihan. Akibatnya model kurang baik mendeskripsikan data-data diluar set pelatihan. Salah satu penyebab *overfitting* adalah sedikitnya jumlah data pelatihan terutama untuk model yang memiliki banyak parameter. Penyebab lainnya adalah adanya *noise* pada data.

## 2.6 Teknik Validasi

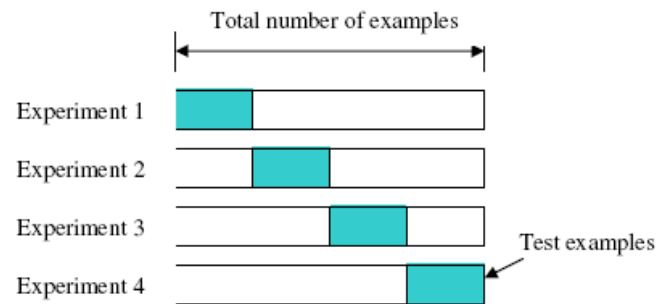
Pada proses pelatihan, beberapa model dibuat dari set pelatihan dimana model tersebut dibuat dengan parameter terbaik dari algoritma yang terpilih. Model ini dibuat dari subset set pelatihan yang disebut set estimasi dimana subset lain



disebut set validasi yang digunakan model untuk menentukan estimasi parameter terbaik

Beberapa parameter seperti nilai error (C), lebar  $\epsilon$  dan fungsi pemetaan harus dipilih dengan hati-hati ketika membangun model SVM. Pemilihan nilai parameter ini sangat menentukan kinerja model SVM (Chen et al., 2004) dan mempengaruhi hasil prediksi. Oleh karena itu, untuk mendapatkan parameter yang sesuai, *n-fold cross validation* biasanya digunakan oleh para peneliti. Sebagai contoh Chen et al. (2004) pada studi mereka menggunakan *n-fold cross validation* dengan membagi secara acak data ke dalam sepuluh set dan menggunakan masing-masing set sebagai set validasi untuk melakukan pelatihan.

Metode *n-fold cross validation* melakukan perulangan  $n$  kali untuk sampel set yang dibagi secara acak ke dalam  $k$  subset yang saling terpisah dimana satu subset digunakan untuk pengujian dan yang lainnya digunakan untuk pelatihan. Proses ini berulang hingga semua subset diuji. Metode ini sama baiknya bila digunakan pada sampel berukuran besar maupun kecil.



Gambar 2.13 Metode *n-fold cross validation*. Untuk masing-masing eksperimen  $n$ , menggunakan subset  $n-1$  untuk proses pelatihan dan subset berikutnya untuk pengujian

## 2.7 Perhitungan Error

Dalam sebuah peramalan selalu dibutuhkan pengukuran untuk mengetahui tingkat kesalahan yang terjadi saat peramalan. Kesalahan (error) ini merupakan penyimpangan variabel yang diukur dari harga sebenarnya. Ketelitian dan ketepatan merupakan dua faktor penting yang menentukan tingkat keberhasilan suatu peramalan.

Ada beberapa teknik pengukuran error yang sering digunakan, seperti *Mean Absolute Error* (MAE) (Lu et al., 2002), *Mean Squared Error* (MSE) dan *Mean Average Prediction Error* (MAPE) (Chen et al., 2004), *Relative Mean Error* (RME) dan *Root Mean Squared Error* (RMSE) (Wu et al., 2004) saat melakukan pengukuran terhadap kesalahan peramalan.

Di antara beberapa tipe pengukuran error, MSE dan RMSE telah digunakan secara luas dalam berbagai penelitian seperti oleh Valverde et al. (2004) pada *Rainfall Prediction* dan Chen et al. (2004) pada *Load Forecasting*.

Dalam Lung (2006), Bowerman dan O'Connell mendefinisikan MSE sebagai nilai rata-rata kuadrat error dari semua nilai peramalan. Perbedaan mendasar antara MSE dengan pengukuran error yang lain adalah MSE memberikan nilai yang lebih untuk tingkat kesalahan yang lebih besar dibanding tingkat kesalahan yang lebih kecil. Oleh karena itu, semakin kecil nilai error peramalan, semakin kecil pula nilai MSE. Persamaan untuk perhitungan MSE adalah sebagai berikut.

$$MSE = \frac{\sum_{i=1}^n (a_i - p_i)^2}{n} \quad (2.23)$$

Dimana

$a_i$  = nilai aktual pada waktu  $i$

$p_i$  = nilai prediksi pada waktu  $i$

$n$  = jumlah data

Untuk mendapatkan tingkat akurasi peramalan juga digunakan *Mean Absolute Percentage Error* (MAPE). MAPE merupakan pengukuran akurasi dalam sebuah data *time series*. MAPE biasanya ditunjukkan dalam nilai persentase. Penggunaan nilai persentase membuat ukuran error menjadi independen. Oleh karena itu perbandingan error pada data yang berbeda memungkinkan untuk dilakukan (Anggraini, 2009). Persamaan untuk perhitungan MAPE adalah sebagai berikut.

$$MAPE = \sum_{i=1}^n \frac{\left| \frac{a_i - p_i}{a_i} \right|}{n} \quad (2.24)$$

Dalam Nor et al. (2007), Jonshon dan King menyatakan bahwa nilai MAPE di bawah 30% menunjukkan nilai prediksi bisa dipertimbangkan tingkat keakurasiannya. Selanjutnya prediksi bisa dikatakan sangat baik apabila nilai MAPE kurang dari 10% (Nor et al., 2007).

## **2.8 Unified Modelling Language (UML)**

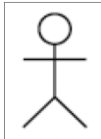

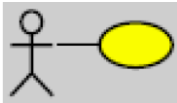
Booch et al. (2005) mendefinisikan UML sebagai bahasa standar untuk menuliskan cetak biru perangkat lunak. UML digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan bagian dari suatu sistem perangkat lunak. UML cocok untuk memodelkan sistem mulai dari sistem informasi enterprise hingga aplikasi web terdistribusi, bahkan juga *embedded system* yang bersifat *hard real time*. UML merupakan bahasa yang ekspresif dan dapat menangani semua sudut pandang yang dibutuhkan untuk membangun sistem-sistem tersebut. Walaupun demikian, UML tetap mudah dipahami dan digunakan.

Terdapat tujuh jenis diagram dalam UML (Schmidt et al., 1999). Diagram tersebut diantaranya yaitu *Use Case Diagrams*, *Interaction Diagrams*, *State Diagrams*, *Activity Diagrams*, *Package Diagrams*, dan *Deployment Diagrams*. Untuk memvisualisasi, menspesifikasi, dan mendokumentasi bagian-bagian sistem perangkat lunak tidak harus menggunakan semua diagram tersebut di atas. Akan dijelaskan *Use Case Diagrams*, *Class Diagrams* dan *Activity Diagrams*.

### **2.8.1 Use Case Diagrams**

*Use case diagrams* merepresentasikan skenario dari sistem perangkat lunak sehingga dapat membantu dalam memahami *requirement* dari sistem perangkat lunak (Schmidt et al., 1999). Terdapat beberapa simbol yang digunakan dalam *Use case diagrams*, dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol dalam *Use Case Diagrams*

| No. | Simbol  | Nama         | Keterangan   |
|-----|---|--------------|--|
| 1.  |    | <i>Actor</i> | <p><i>Actor</i> adalah suatu peran dalam skenario sistem perangkat lunak, <i>actor</i> tidak harus manusia ataupun <i>user</i> tetapi juga bisa sistem eksternal.</p> <p>Contoh Actor : <i>Trading manager, Tradet, Salesperson, Accounting System.</i></p> <p><i>Actor single</i> dapat merepresentasikan <i>multiple user</i> atau sistem dan <i>single user</i>, keduanya dapat memainkan banyak peran.</p> |
| 2.  |  | <i>Case</i>  | <i>Case</i> merepresentasikan aksi atau peran dari <i>actor</i> .  |
| 3.  |  | <i>Line</i>  | Antara <i>actor</i> dan <i>case</i> dihubungkan oleh garis. Arti dari garis tersebut, <i>actor</i> berperan <i>case</i> .  |

### 2.8.2 Class Diagram

*Class Diagram* menjelaskan objek-objek yang terdapat dalam sistem dan berbagai hubungan statis yang terjadi diantara objek-objek tersebut. *Class diagram* merepresentasikan atribut dan operasi dalam suatu *class* dan batasan-batasan hubungan yang terjadi antara objek satu dengan yang lainnya. *Class Diagram* adalah pusat untuk menganalisis, desain, dan implementasi (Scmidt et al., 1999). Beberapa simbol yang digunakan dalam *class diagram* dapat dilihat pada tabel 2.4. Terdapat tiga pendekatan dalam *class diagram*, yaitu :

1. *Conceptual*

Merepresentasikan konsep hubungan antara *class* satu dengan *class* yang lain, tidak terikat dengan bahasa yang digunakan dalam implementasi, *language independence*.

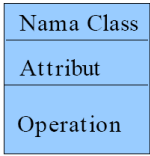

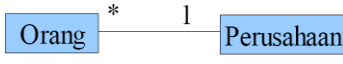


2. *Spesification*

Merepresentasikan antar muka (*software interface*).

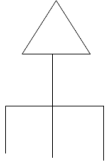
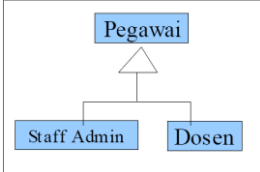
3. *Implementation*

Merepresentasikan *class* yang benar-benar digunakan dan diimplementasikan menggunakan bahasa pemrograman.

Tabel 2.4 Simbol dalam *Class Diagrams*

| No. | Simbol  | Nama                                | Keterangan  |
|-----|---|-------------------------------------|---|
| 1.  |    | <i>Class</i>                        | Bagian atas menunjukkan nama <i>class</i> , atribut dari <i>class</i> tersebut dan diikuti oleh operasi dari <i>class</i> . <i>Operation</i> sering disebut juga <i>method</i> pada implementasinya.  |
| 2.  |  | <i>Association : Multiplicities</i> | <i>Multiplicities</i> mengindikasikan banyaknya objek yang terlibat dalam suatu hubungan. Contohnya : Seorang hanya dapat bekerja pada satu perusahaan, sedangkan satu perusahaan dapat mempekerjakan lebih dari satu orang. Maka hubungan antara class orang dan perusahaan  |
| 3.  |  | <i>Aggregation</i>                  | <i>Aggregation</i> merepresentasikan hubungan yang menyatakan bagian dari, CPU bagian dari Computer   |



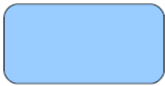

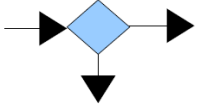
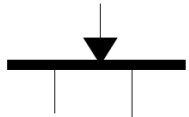


|    |   |                |  |
|----|---|----------------|--|
| 4. |  | Generalization | <p>Apabila beberapa <i>class</i> memiliki <i>superclass</i> yang sama. <i>Class</i> staff administrasi dan dosen memiliki <i>superclass</i> yang sama yaitu <i>class</i> pegawai.</p>  |
|----|---|----------------|--|

### 2.8.3 Activity Diagram

*Activity diagrams* merepresentasikan *workflow* dari seluruh proses yang terjadi dalam sistem. Perbedaan dengan *flowchart*, *activity diagram* dapat merepresentasikan proses yang terjadi secara paralel atau bersamaan (Schmidt et al., 1999).

Tabel 2.5 Simbol dalam *Activity Diagrams*

| No. | Simbol  | Nama                       | Keterangan   |
|-----|---|----------------------------|--|
| 1.  |    | <i>Start</i>               | Simbol untuk memulai suatu proses  |
| 2.  |    | <i>End</i>                 | Akhir proses   |
| 3.  |    | <i>Activity</i>            | Menyatakan aktivitas dalam suatu proses.   |
| 4.  |   | <i>Trigger</i>             | Menyatakan <i>activity</i> satu merupakan triger bagi <i>activity</i> selanjutnya.                               |
| 5.  |  | <i>Decision Activity</i>   | Untuk menyatakan percabangan, <i>nested decision</i>   |
| 6.  |  | <i>Synchronization Bar</i> | <i>Synchronozation bar</i> digunakan untuk menyatakan <i>trigger</i> yang terjadi secara bersamaan atau paralel. |

## **BAB III**

### **ANALISIS DAN RANCANGAN SISTEM**

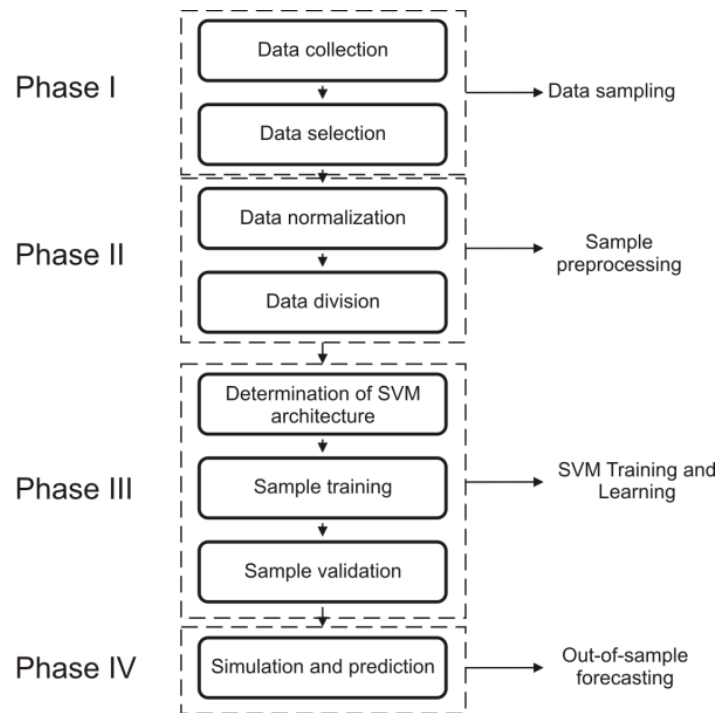
#### **3.1 Analisis Sistem**

Regresi merupakan studi mengenai perhitungan satu variabel berdasarkan nilai dari variabel lain yang mempunyai kaitan dengan variabel yang akan diprediksi. Prediksi dilakukan dengan memperhitungkan variabel-variabel tersebut sehingga membentuk suatu pola. Pola inilah yang nantinya akan digunakan dalam melakukan prediksi.

Dalam perkembangannya, permasalahan regresi diselesaikan mulai dari penggunaan model *time series prediction* hingga *artificial neural network*. Namun metode-metode tersebut memiliki beberapa kelemahan yang sangat berpengaruh terhadap proses prediksi dan untuk menyelesaikan berbagai permasalahan tersebut dikembangkanlah metode *Support Vector Machine*.

Pada tugas akhir ini, akan dibangun sebuah prototipe perangkat lunak untuk menyelesaikan permasalahan regresi dengan menggunakan metode SVM kemudian akan dilakukan pengujian untuk mengetahui kinerja prototipe perangkat lunak tersebut dengan perhitungan *Mean Squared Error*. Untuk selanjutnya, prototipe perangkat lunak yang dibangun dalam tugas akhir ini disebut dengan SVReg.

Secara umum diagram alir sistem yang akan dikerjakan ditunjukkan pada gambar di bawah ini.



Gambar 3.1 Diagram alir sistem

Ada dua hal penting yang harus diperhatikan dalam SVReg, yaitu data dan sistem itu sendiri. Sebelum data dimasukkan ke dalam sistem yang mengimplementasikan algoritma, data harus dikumpulkan, diteliti, dibersihkan dan dipilih. Sistem yang baik sekalipun akan gagal melakukan proses prediksi apabila ada kesalahan dalam analisa data yang digunakan. Dalam tugas akhir ini, model SVM dibangun menggunakan *Support Vector Regression* yang sudah banyak digunakan dalam prediksi *time series*. Secara umum data yang sudah dikumpulkan dan dianalisa akan ditransformasikan ke dalam format SVM yang banyak digunakan dalam berbagai penelitian.

Langkah pertama pembangunan sistem ini adalah membentuk suatu model arsitektur yang nantinya akan digunakan sebagai acuan dalam melakukan prediksi. Jika dalam permasalahan regresi linear, eksponensial, maupun polinomial model ini berupa suatu persamaan yang membentuk suatu garis, maka dalam SVM model ini berupa suatu bidang dan *support vector*. Model SVM dibangun melalui suatu proses pelatihan seperti yang dilakukan pada jaringan syaraf tiruan. Proses pelatihan ini dilakukan guna menentukan *support vector* yang berpengaruh dalam pembentukan bidang pemisah. Pada proses pelatihan ini ada beberapa parameter yang perlu diperhatikan, seperti fungsi kernel, jenis SVM, parameter C dan epsilon. Penggunaan *n-cross validation* digunakan untuk mengetahui sejauh mana kinerja SVReg yang ditunjukkan dengan nilai MSE. *n-cross validation* ini juga digunakan untuk menentukan parameter terbaik yang digunakan untuk membangun model SVM.

Karena belum ada cara yang terstruktur untuk mendapatkan parameter terbaik dalam pembangunan model SVM, maka parameter kernel yang menghasilkan parameter terbaik dalam set validasi digunakan dalam pembangunan model. Pemilihan parameter ini harus diperhatikan dengan seksama karena mempunyai pengaruh yang cukup besar dalam pembangunan model. Untuk memastikan bahwa parameter yang digunakan adalah parameter terbaik, maka *cross validation* digunakan untuk mengevaluasi model dengan beberapa parameter yang berbeda dan nilai *Mean Square Error* akan dievaluasi.

Seperti pada permasalahan regresi secara umum, dibutuhkan data *time series* untuk membentuk model SVM. Secara umum, data ini disimpan dalam bentuk file

.txt dengan format tertentu. Data ini tidak disimpan dalam database. Data ini dikategorikan ke dalam dua data set, yaitu data set pelatihan yang digunakan dalam proses pelatihan dan data set prediksi, yaitu data yang digunakan dalam melakukan prediksi. Sedangkan hasil pelatihan tersimpan ke dalam bentuk file .model. File output hasil prediksi juga akan tersimpan dalam bentuk file .output.

Perancangan sistem yang baik harus mempertimbangkan kemampuan komponen-komponen sistem tersebut untuk digunakan pada sistem yang lain. Analisis dan perancangan yang berorientasi objek memungkinkan hal tersebut. SVReg menggunakan analisis dan perancangan berorientasi objek dimana salah satu tujuan yang ingin dicapai dalam pengembangan perangkat lunak berorientasi objek adalah *reusability*, sehingga pengembangan lebih lanjut terhadap sistem yang ada akan lebih mudah dilakukan. Oleh karena itu beberapa bagian dari sistem ini akan mengadopsi *class library* yang sudah ada. Sesuai dengan batasan masalah dalam penyusunan tugas akhir ini, SVReg menggunakan *class library* libsvm (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>) .

### 3.2 Spesifikasi Sistem

Berdasarkan analisis yang telah dilakukan, sistem ini akan memiliki spesifikasi sebagai berikut.

1. Mampu membaca (melakukan *parsing*) terhadap file input (data set pelatihan maupun data set prediksi) dengan format :

```
<label> <index1>:<value1> <index2>:<value2> ...
```

`<label>` dan `<value>` adalah nilai real. `<index>` adalah nilai integer yang dimulai dari 1. `<label>` pada data set prediction digunakan untuk perhitungan error. Jika nilainya tidak diketahui, bisa diisi dengan 0 atau nilai sembarang.

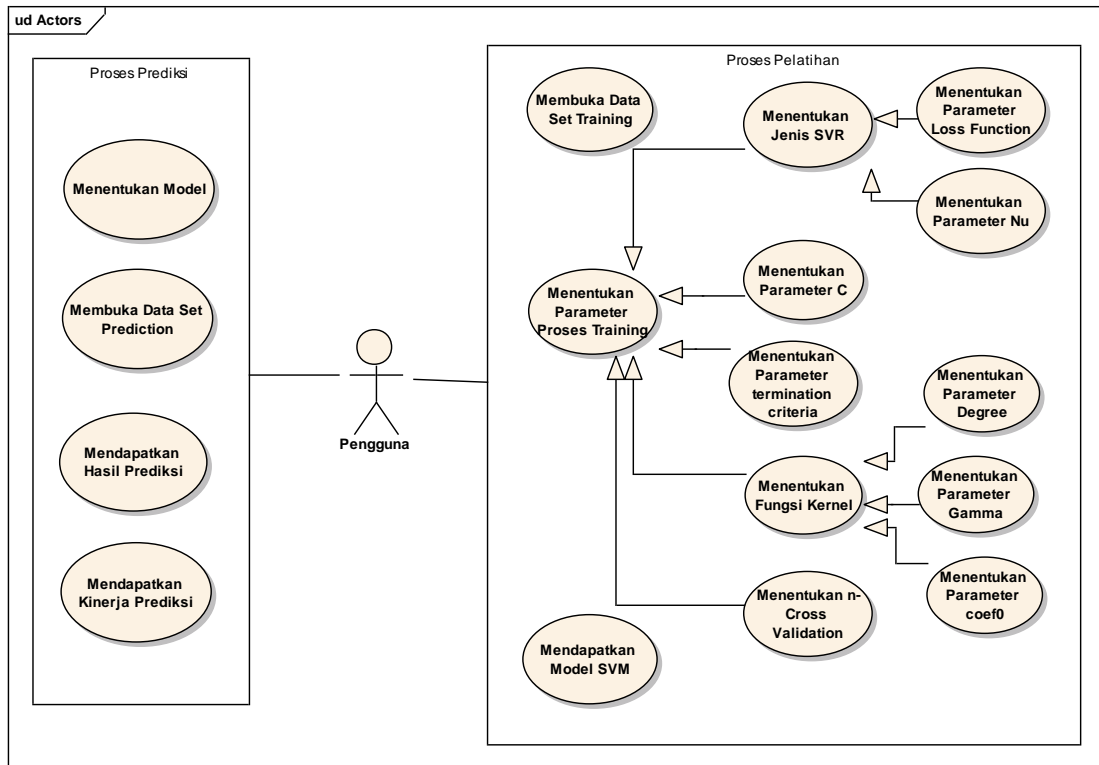
2. Dapat diatur parameter dalam proses pelatihannya yang meliputi jenis SVM (epsilon SVR atau nu SVR), nilai parameter C, nilai epsilon, fungsi kernel (linear, polinomial, radial basis function atau sigmoid) dan *n-cross validation*.
3. Mampu melakukan proses pelatihan untuk mendapatkan model SVM yang nantinya digunakan untuk melakukan prediksi dan langsung tersimpan dalam file berformat .model.
4. Mampu melakukan prediksi berdasarkan data-data sebelumnya.
5. Mampu melakukan perhitungan.
6. Data set training, data set prediction dan model tidak disimpan dalam database, akan tetapi disimpan dalam bentuk file.
7. Antarmuka pengguna dan class library terpisah.

### 3.3 Rancangan Sistem

#### 3.3.1 Use Case Diagram

UML menyediakan *use case diagram* untuk menggambarkan spesifikasi kebutuhan perangkat lunak. Spesifikasi perangkat lunak dengan kata-kata saja bersifat ambigu dan sulit untuk dipahami. *Use case diagram* untuk sistem ini hanya memiliki satu aktor bernama "Pengguna". Terdapat 17 *use case*. Semua *use case*

tersebut diturunkan dari spesifikasi sistem yang telah dijabarkan pada bagian 3.2. *Use case diagram* untuk sistem ini secara lengkap ditunjukkan pada gambar 3.2.



Gambar 3.2. *Use Case Diagram*

Secara umum, *use case diagram* pada gambar 3.2. bisa dibagi menjadi dua proses, yaitu proses pelatihan dan proses prediksi. Penjelasan *use case diagram* pada gambar di atas.

Proses pelatihan :

1. *Use Case Membuka File Data Set Training*

Menentukan file yang akan dijadikan sebagai input data set pelatihan yang akan digunakan dalam proses pelatihan SVReg.

2. *Use Case Menentukan Parameter Proses Pelatihan*



Menentukan parameter-parameter yang akan digunakan dalam proses pelatihan, meliputi penentuan fungsi kernel, jenis SVM, nilai parameter  $C$ , parameter  $\epsilon$  dan  $n$ -cross validation.

### 3. Use Case Menentukan Jenis SVM

Menentukan jenis SVM yang akan digunakan dalam proses pelatihan, pengguna dapat memilih epsilon SVR atau nu SVR. Pada dasarnya epsilon SVR dan nu SVR sama, perbedaannya hanya pada parameternya. Nilai epsilon pada epsilon SVR berkisar antara 0 sampai tak hingga sedangkan nu selalu bernilai antara 0 dan 1.

### 4. Use Case Menentukan Parameter Epsilon

Menentukan nilai  $\epsilon$  pada epsilon SVR. Radius  $\epsilon$  menentukan data di dalam  $\epsilon$  tube yang diabaikan dalam regresi. Nilai  $\epsilon$  sebaiknya diset pada angka yang kecil.

### 5. Use Case Menentukan Parameter nu

Menentukan nilai nu yang membatasi nilai *upper bound* dari *training error* dan *lower bound* pada fraksi data point yang menjadi *Support Vector*.

### 6. Use Case Menentukan Parameter Termination Criteria

Menentukan *stopping criteria* proses pelatihan.

### 7. Use Case Menentukan Parameter C

Menentukan nilai parameter  $C$ .  $C$  adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna.

#### 8. *Use Case Menentukan Fungsi Kernel*

Menentukan fungsi kernel yang akan digunakan dalam proses pelatihan, pengguna dapat memilih fungsi kernel Linear, Polinomial, Radial Basis maupun Sigmoid. Masing-masing fungsi kernel ini nantinya berkaitan dengan pengaturan nilai parameter *gamma*, *coefisien* dan *degree*.

#### 9. *Use Case Menentukan Parameter Degree*

Parameter *degree* digunakan dalam fungsi kernel Polynomial.

#### 10. *Use Case Menentukan Parameter Gamma*

Parameter *gamma* digunakan dalam fungsi kernel Polinomial, Radial Basis, dan Sigmoid. Nilai *gamma* harus lebih besar dari 0.

#### 11. *Use Case Menentukan Parameter Coef0*

Parameter *coefisien* digunakan dalam fungsi kernel Polinomial dan Sigmoid.

#### 12. *Use Case Menentukan n-cross validation*

Menentukan jumlah set validasi yang digunakan dalam proses validasi untuk menentukan parameter terbaik yang digunakan untuk membangun model SVM.

#### 13. *Use Case Mendapatkan Model SVM*

Pengguna mendapatkan hasil dari proses pelatihan, yakni model SVM. Model ini disimpan dalam bentuk file .model yang nantinya digunakan dalam proses prediksi.

## Proses Prediksi

### 1. *Use Case* Membuka File *Data Set Prediction*

Menentukan file yang akan dijadikan sebagai input data set prediksi yang digunakan dalam proses prediksi SVReg.

### 2. *Use Case* Menentukan Model SVM

Menentukan file model yang akan dijadikan sebagai model SVM dalam melakukan proses prediksi. File model ini berekstensi .model.

### 3. *Use Case* Mendapatkan Hasil Prediksi

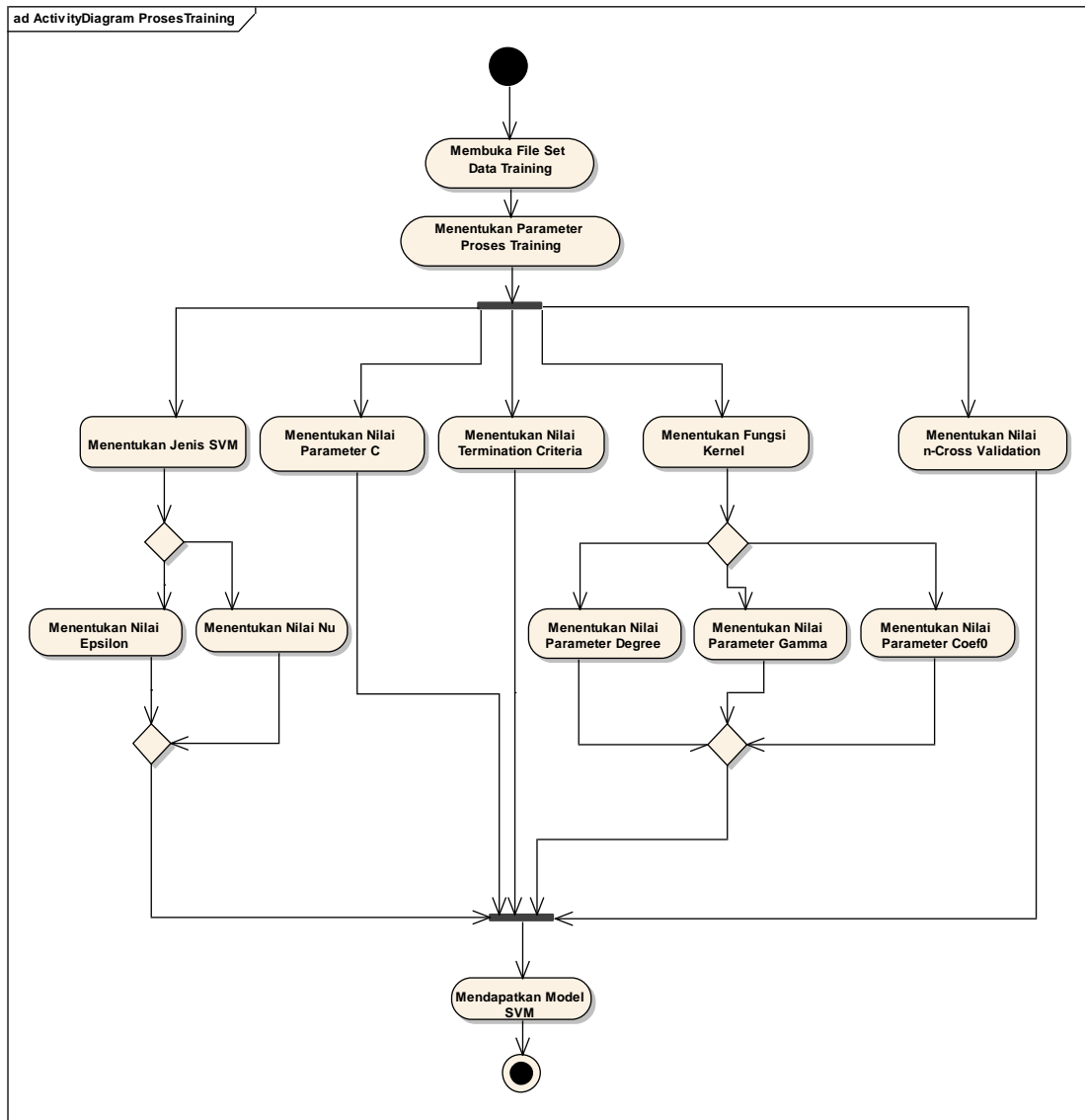
Pengguna mendapatkan hasil prediksi dari *data set prediction* sebelumnya. Prediksi ini didapatkan melalui proses regresi. File disimpan dalam format .output dan disimpan dalam direktori yang sama dengan file *data set prediction*.

### 4. *Use Case* Mendapatkan Kinerja Sistem

Pengguna mendapatkan kinerja sistem yaitu berupa nilai *Mean Squared Error* dan *Mean Absolute Percentage Error*.

## 3.3.2 Activity Diagram

Sistem ini dapat digambarkan secara sederhana menggunakan *activity diagram*. *Activity diagram* sistem ini dapat dibagi menjadi dua, yaitu proses pelatihan dan proses prediksi. Untuk *activity diagram* proses training dapat dilihat pada gambar

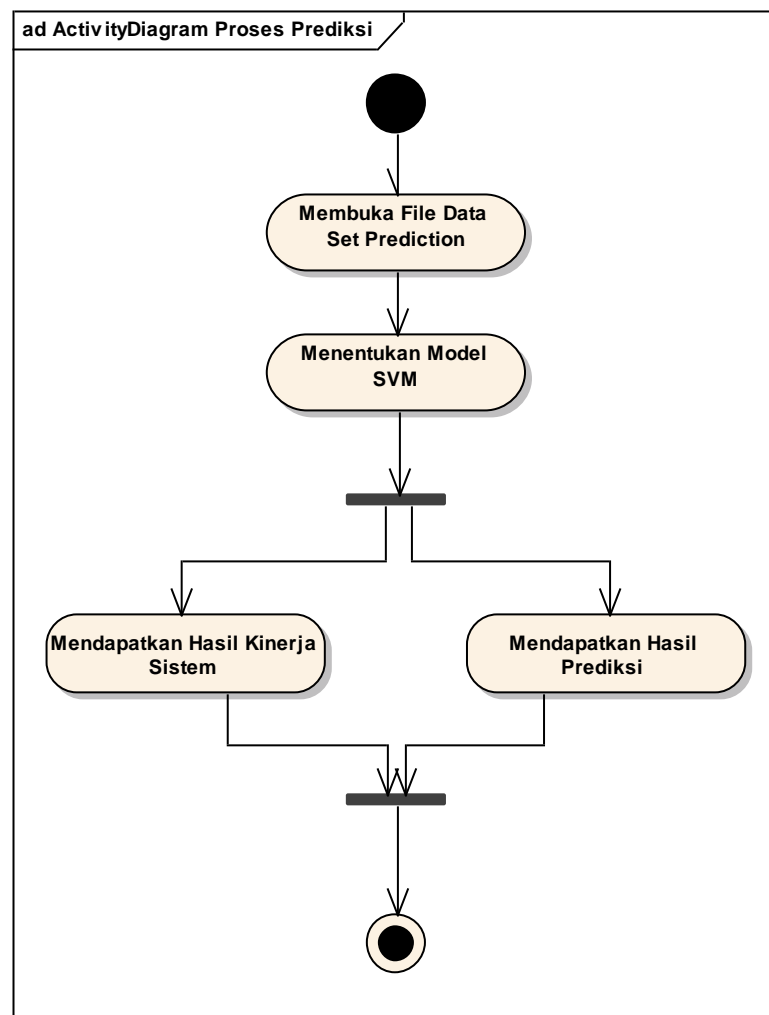


Gambar 3.3 Activity diagram proses pelatihan

Pada proses pelatihan, pengguna membuka input file data set training, selanjutnya menentukan parameter proses pelatihan. Ada beberapa parameter yang harus ditentukan oleh pengguna, yaitu jenis SVM, nilai parameter  $C$ , *termination criteria*, fungsi kernel dan nilai *n-cross validation*. Ada dua tipe SVM yang bisa digunakan, yakni epsilon SVR dan nu SVR. Masing-masing mempunyai parameter

tersendiri, yakni epsilon untuk epsilon SVR dan nu untuk nu SVR. Tiap-tiap fungsi kernel memiliki parameter tersendiri, yakni parameter gamma, degree, dan coefisien. Selanjutnya sistem akan melakukan proses pelatihan dan dari hasil pelatihan ini didapatkan model SVM.

*Activity diagram* untuk proses prediksi dapat dilihat pada gambar 3.4.



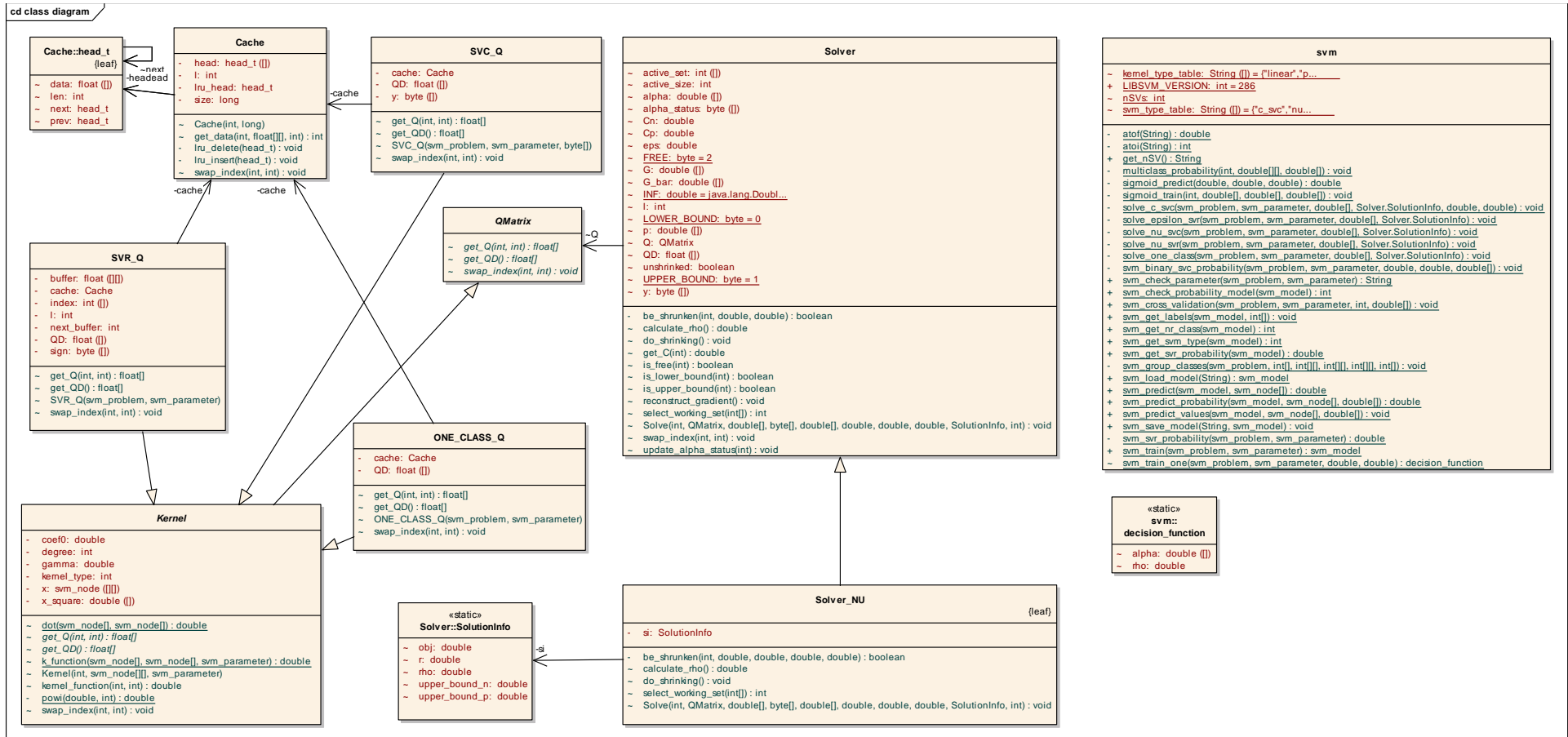
Gambar 3.4 *Activity diagram* proses prediksi

Pada proses prediksi, pengguna membuka input file data set prediction. Selanjutnya pengguna menentukan model SVM yang akan digunakan untuk

melakukan prediksi. Model SVM didapatkan dari proses pelatihan yang dilakukan sebelumnya. Sistem akan melakukan prediksi dan menampilkan hasil prediksi dan hasil kinerja sistem kepada pengguna.

### 3.3.3 Class Diagram

Sesuai dengan batasan masalah dalam penyusunan tugas akhir ini, SVReg menggunakan *class library* libsvm. Terdapat beberapa bagian yang tidak dibutuhkan, maka *class library* ini kemudian dipaket ulang dengan hanya menyertakan bagian-bagian *class library* yang diperlukan bagi SVReg.



Gambar 3.5 Class diagram libsvm

*Class diagram* pada gambar di atas adalah *class diagram* library libsvm.

Beberapa *class* yang tertera dalam *class diagram* akan dijelaskan.

1. *Class Cache*

*Class* ini berisi implementasi *caching* yang digunakan untuk mempercepat proses komputasi dengan fungsi kernel.

2. *Class Kernel*

*Class* ini berisi implementasi berbagai macam fungsi kernel seperti Linear, Polynomial, RBF dan Sigmoid.

3. *Class Solver*

*Class* ini berisi algoritma SMO yang telah dimodifikasi.

4. *Class svm*

*Class* ini menyediakan berbagai macam *method* yang akan digunakan dalam proses training dan proses prediksi SVReg. Seperti *method* `svm_save_model` yang digunakan untuk menyimpan model SVM saat proses pelatihan. *Method* `svm_load_model` yang digunakan untuk me-load model SVM saat dilakukan prediksi. *Method* `svm_check_parameter` untuk mengecek kesesuaian parameter input dari pengguna terhadap sistem. *Method* `svm_train` dan *method* `svm_predict` yang sangat penting dalam proses pelatihan dan prediksi serta *method* `svm_cross_validation` untuk mendapatkan nilai MSE.



Berdasarkan *class-class* tersebut di atas maka setiap langkah dalam SVReg akan menggunakan *class* sebagai berikut.

1. Proses pelatihan untuk mendapatkan parameter terbaik

Proses training dilakukan untuk mendapatkan parameter terbaik yang didasarkan pada nilai MSE terkecil. *Method* `svm_check_parameter` akan mengecek apakah nilai parameter masukan dari pengguna memenuhi kriteria `libsvm`. *Method* ini harus dipanggil sebelum memanggil *method* `svm_cross_validation` dan `svm_train`. Untuk mendapatkan nilai MSE ini digunakan *cross validation* yang dilakukan oleh *method* `svm_cross_validation` dari *class* `svm`.

2. Menyimpan model

Setelah didapatkan kombinasi parameter terbaik berdasarkan nilai MSE terendah, data set training akan dilatih dengan menggunakan *method* `svm_train` dari *class* `svm` yang akan menyimpan model SVM yang dibangun berdasarkan kombinasi parameter terbaik dengan menggunakan *method* `svm_save_model` yang ada pada *class* `svm`.

3. Proses prediksi

Proses prediksi menggunakan *method* `svm_load_model` untuk me-load model SVM yang telah dibangun dan disimpan dalam proses pelatihan dan *method* `svm_predict` akan menangani proses prediksi.

### 3.3.4 Rancangan Antar Muka

Terdapat dua form utama dalam sistem ini, yaitu Form Training dan Form Prediction. Terdapat satu menu utama yang terdiri dari tiga item, yaitu :

1. About, yang berisi penjelasan singkat mengenai SVReg.
2. Help, yang berisi penjelasan singkat mengenai pengaturan nilai parameter.
3. Exit, untuk keluar dari program.

#### 3.3.4.1. Form Training

Gambar 3.6 Form training

Pada Form Training, pengguna melakukan proses pelatihan terhadap sistem. Di sini pengguna dapat menentukan file input yang digunakan dengan format yang telah ditentukan sebelumnya, menentukan jenis SVM yang digunakan, menentukan nilai *termination criteria*, menentukan fungsi kernel yang digunakan dalam

melakukan pemetaan terhadap file input serta menentukan nilai *n-cross validation* yang digunakan dalam proses validasi.

Ada dua *drop down list*, yaitu SVM Type dan Kernel Function. *Drop down list* SVM Type berisi dua jenis SVM yang digunakan, yaitu epsilon SVR dan nu SVR. Pada epsilon SVR, pengguna menentukan nilai parameter C, epsilon dan *termination criteria*. Sedangkan pada nu SVR, pengguna menentukan nilai parameter C, nu dan *termination criteria*.

*Drop down list* Kernel Function digunakan untuk memilih fungsi kernel yang digunakan dalam proses pelatihan. Saat menentukan fungsi kernel, pengguna juga dapat menentukan nilai parameter degree, coefisien, dan gamma untuk fungsi kernel polynomial; nilai parameter gamma untuk fungsi kernel Radial Basis Function; serta nilai parameter gamma dan coefisien untuk fungsi kernel Sigmoid. Sedangkan untuk fungsi kernel Linear tidak ada parameter yang perlu ditentukan oleh pengguna.

Pada form ini juga terdapat tiga *button*, yaitu Data Set Training, Train dan Reset. *Button* Data Set Training digunakan untuk memilih file input yang akan digunakan sebagai data set dalam proses pelatihan. *Button* Train digunakan untuk melakukan proses pelatihan sistem, sedangkan *button* Reset digunakan untuk melakukan reset terhadap proses pelatihan sistem. Pada *text area* Data Set Training, pengguna dapat melihat data-data yang akan digunakan dalam proses pelatihan. Sedangkan pada *text area* Model, pengguna dapat melihat hasil evaluasi dari proses pelatihan.

### 3.3.4.2. Form Prediction

Gambar 3.7 Form prediction

Form Prediction ini digunakan oleh pengguna untuk melakukan prediksi. Pada form ini terdapat empat *button*, yaitu Data Set Prediction, Model, Predict, dan Reset. *Button* Data Set Prediction digunakan pengguna untuk menentukan data set yang akan digunakan untuk melakukan prediksi. Prediksi terhadap satu waktu membutuhkan data *time series* sebelumnya dengan format yang sama seperti data set training. *Button* Model digunakan untuk memilih model SVM mana yang akan digunakan untuk melakukan prediksi. File model ini diperoleh dari proses pelatihan yang dilakukan sebelumnya. *Button* Predict digunakan untuk melakukan proses prediksi, sedangkan *button* Reset digunakan untuk melakukan reset terhadap proses prediksi.

Form ini juga memiliki tiga *text area*, yaitu *text area* Data Set Prediction, Prediction, dan Performance. *Text area* Data Set Prediction berisi data set prediksi, berisi data-data yang digunakan untuk melakukan proses prediksi. *Text area* Prediction digunakan untuk menampilkan hasil prediksi dan *text area* Performance digunakan untuk menampilkan kinerja sistem yang ditunjukkan dengan nilai *Mean Squared Error* dan *Mean Absolute Percentage Error*.

### 3.4 Rancangan Pengujian

SVReg merupakan suatu sistem yang mengaplikasikan *Support Vector Machine* sebagai metode untuk melakukan regresi. Dari regresi ini nantinya didapatkan suatu nilai yang disebut prediksi. Kinerja SVReg perlu diuji untuk menunjukkan bahwa SVReg merupakan salah satu metode yang cukup baik dalam melakukan prediksi. Seperti telah diketahui sebelumnya, SVM telah menghasilkan kinerja yang baik dalam beberapa penelitian *time series prediction*, seperti pada *Stock Price Forecasting* (Bao et al., 2004), *Traffic Speed Prediction* (Vanajakshi and Rilett, 2004) dan *Travel Time Series Prediction* (Wu et al., 2004), serta untuk menunjukkan bahwa SVReg merupakan salah satu alternatif yang baik sebagai pengganti jaringan syaraf tiruan yang mempunyai beberapa kekurangan yaitu ketergantungannya pada struktur jaringan dan kompleksitas sampel yang mengakibatkan *overfitting* dan tingkat generalisasi yang rendah (Kim and Valdes, 2003; Zhu et al., 2002).

Data yang digunakan dalam pengujian terhadap SVReg adalah data harga minyak mentah dunia dan data nilai tukar Rupiah terhadap US Dollar. Data harga minyak mentah dunia berdasarkan West Texas Intermediate Crude Oil (WTI) <http://www.uprr.com/customers/surcharge/wti.shtml> yang tercatat pada tahun 2009 mulai dari tanggal 2 Januari hingga 29 Oktober. Data harga minyak ini merupakan harga penutupan tiap harinya dan hanya menggunakan data lima hari kerja dari Senin hingga Jumat karena WTI tidak mengeluarkan data harga minyak saat akhir pekan.

Data nilai tukar Rupiah terhadap US Dollar didapatkan dari <http://oanda.com> yang tercatat pada tahun 2009 mulai dari tanggal 1 Januari hingga 31 Oktober. SVReg akan melakukan prediksi harga harian, sehingga data yang digunakan adalah data harga harian. Data nilai tukar Rupiah yang digunakan di sini merupakan nilai rata-rata harian yang tercatat pada website OANDA.

Parameter yang dipilih pada proses pelatihan untuk membentuk model SVM yang digunakan dalam proses prediksi dilakukan dengan metode trial and error. Pemilihan parameter akan mempengaruhi kinerja SVM model dalam melakukan prediksi. Beberapa parameter penting yang berkaitan dengan *Support Vector Machine* antara lain nilai C, epsilon dan fungsi kernel.

Pemilihan nilai C dan epsilon mempunyai peranan yang penting dalam membentuk model SVM. Nilai C menentukan hukuman untuk kesalahan estimasi. Semakin besar nilai C akan memberikan hukuman yang lebih tinggi untuk kesalahan estimasi yang pada akhirnya akan meminimalkan error dengan generalisasi yang

lebih rendah dan begitu pula untuk nilai  $C$  yang lebih kecil. Jika nilai  $C$  mendekati tak hingga, model SVM yang kompleks akan dihasilkan sedemikian rupa sehingga SVM tidak mengijinkan terjadinya kesalahan.

Nilai epsilon mampu mempengaruhi jumlah *support vector* yang didapatkan pada proses pelatihan. *Support vector* inilah yang membentuk model SVM sebagai dasar acuan melakukan prediksi. Semakin besar nilai epsilon, semakin sedikit jumlah *support vector* yang dipilih. Dengan kata lain, semakin besar nilai epsilon, akan menghasilkan estimasi yang cenderung 'flat'.

Pemilihan fungsi kernel merupakan hal yang sangat penting karena memungkinkan perkalian *dot product* pada ruang fitur dimensi tinggi tanpa harus mengetahui transformasi nonlinear. Dalam penentuan fungsi kernel ini sendiri ada beberapa parameter yang harus diperhatikan nilainya agar bisa menghasilkan arsitektur model SVM yang optimal dalam melakukan prediksi, yakni coefisien, degree dan gamma.

Beberapa tipe SVM dengan kombinasi fungsi kernel untuk pemetaan akan digunakan untuk membentuk arsitektur model yang paling baik untuk melakukan prediksi. Nilai *Mean Squared Error* (MSE) akan digunakan sebagai evaluasi untuk menentukan model SVM mana yang akan dibangun. Perhitungan MSE digunakan untuk mendapatkan nilai error pada proses pelatihan yang dilakukan SVReg. Alasan penggunaan MSE karena MSE memberikan nilai yang lebih besar untuk tingkat

kesalahan yang lebih besar dibandingkan tingkat kesalahan yang lebih kecil. Sehingga semakin kecil nilai error peramalan, semakin kecil pula nilai MSE.

Pada proses prediksi, penggunaan *Mean Absolute Percentage Error* (MAPE) sebagai dasar perhitungan error lebih diutamakan daripada penggunaan MSE karena sifat MAPE yang independen sehingga bisa digunakan untuk membandingkan error prediksi pada data set yang berbeda.

Hsu et al. (2003) telah menyarankan sebuah prosedur dalam mengimplementasikan SVM dengan menggunakan libsvm. Dalam studi ini, prosedur ini diimplementasikan dengan sedikit adaptasi menurut tujuan dan ruang lingkup studi untuk menguji kinerja SVReg. Langkah-langkah pengujian dapat dilihat di bawah ini.

Langkah 1 : Tentukan data yang akan digunakan. Jika perlu lakukan normalisasi data.

Langkah 2 : Ubah data ke dalam format libsvm

```
<label> <index1>:<value1> <index2>:<value2> ...
```

Langkah 3 : Membangun model SVM dengan menentukan SVM Type, mengatur nilai parameter C dan epsilon.

Langkah 4 : Menentukan kernel type.

Jika kernel Linear, tidak perlu mengatur parameter apapun.

Jika kernel Polinomial, mengatur nilai parameter degree, gamma dan coef 0.

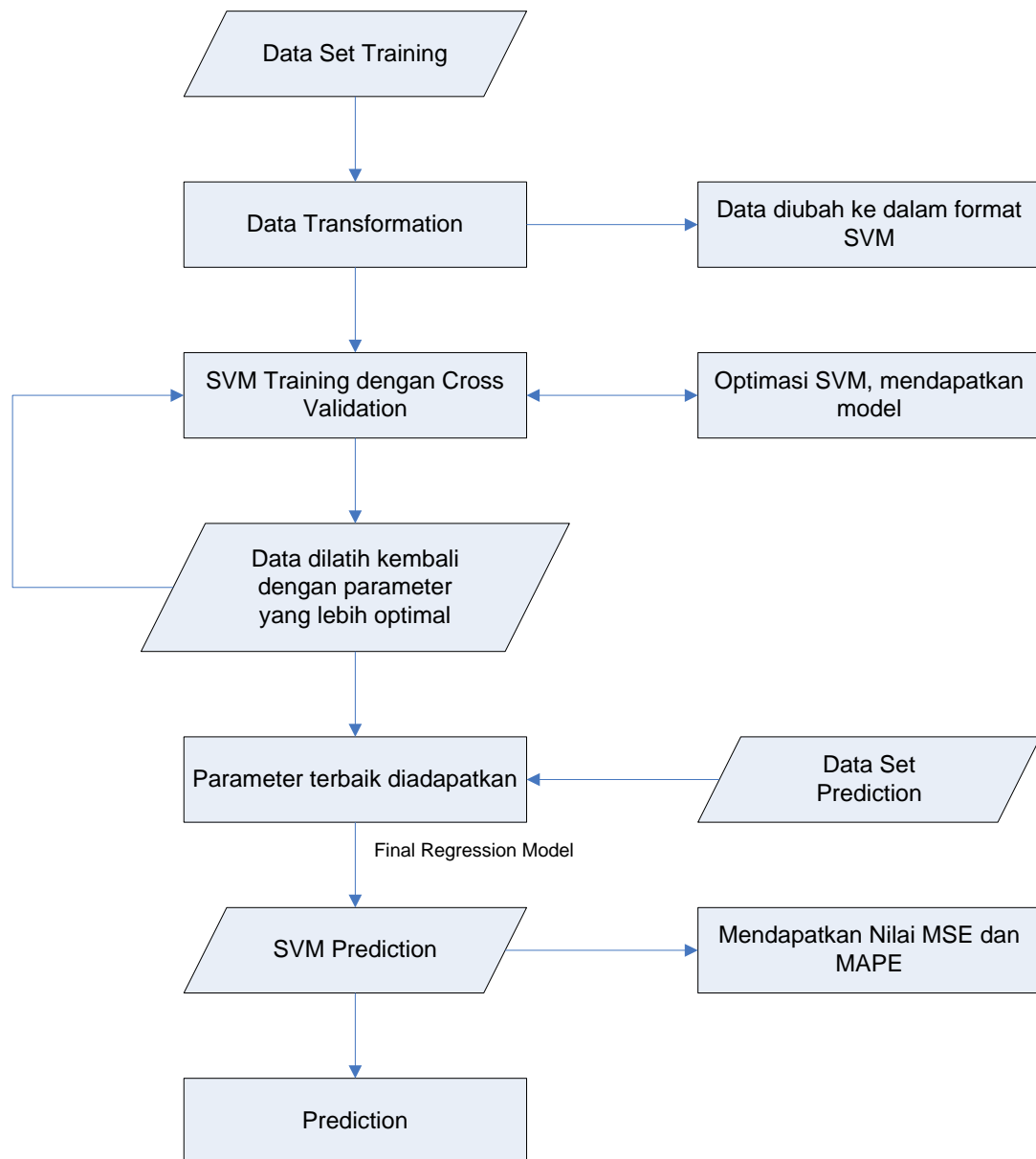
Jika kernel RBF, mengatur nilai parameter gamma.



Jika kernel Sigmoid, mengatur nilai parameter gamma dan coef 0.

- Langkah 5 : Gunakan *cross validation* untuk mendapatkan parameter terbaik. Atur jumlah  $n$  pada *n-cross validation*.
- Langkah 6 : Mulai pelatihan dengan menggunakan input data dan parameter standar.
- Langkah 7 : Tes sistem dengan menggunakan training data dan validation data set untuk menemukan parameter terbaik. Ulangi langkah 3 dan 4 sampai didapatkan nilai MSE terendah.
- Langkah 8 : Ganti nilai parameter  $C$  dan epsilon serta kernel type beserta atributnya dengan nilai parameter yang menghasilkan nilai MSE terendah.
- Langkah 9 : Uji data dengan menggunakan data set training untuk mendapatkan model SVM terbaik.
- Langkah 10 : Pengujian dengan menggunakan data set prediction.
- Langkah 11 : Mulai prediksi dengan menggunakan model SVM terbaik yang didapatkan dalam proses pelatihan.
- Langkah 12 : Hitung nilai error dengan menggunakan MSE dan MAPE.

Untuk mendapatkan gambaran mengenai pengujian SVReg, flowchart langkah-langkah pengujian SVReg dapat dilihat pada gambar 3.9



Gambar 3.8 Flowchart pengujian SVReg

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Pembangunan Sistem**

SVReg dibangun menggunakan platform dan bahasa pemrograman Java. Java dipilih karena beberapa alasan :

1. Memungkinkan aplikasi dikompilasi sekali dan dijalankan di mana saja.
2. Java merupakan bahasa berorientasi objek yang saat ini merupakan paradigma standar dalam rekayasa perangkat lunak.
3. Java juga mengambil alih fungsi manajemen memori dari tangan programmer sehingga pembangunan sistem bisa lebih difokuskan pada penyelesaian masalah utama yang dihadapi.

Beberapa perangkat lunak yang dipakai untuk membangun sistem adalah sebagai berikut.

1. Sistem Operasi Windows 7 Professional
2. Java Development Kit 1.5.0\_01
3. Net Beans 6.5.1

Perangkat keras yang digunakan adalah laptop Compaq dengan spesifikasi sebagai berikut.

1. Prosesor Intel Core 2 Duo
2. Memori DDR2 1 GB
3. Hard Disk 320 GB

## 4.2 Implementasi Algoritma

Sistem SVReg memiliki tiga *class* utama yaitu *svm\_train.java*, *svm\_predict.java* dan *mainInterface.java*. *Class svm\_train.java* berisi implementasi algoritma pelatihan program dengan metode SVM, *class svm\_predict.java* berisi implementasi algoritma prediksi program sedangkan *class mainInterface.java* berisi desain antarmuka dari SVReg.

### 4.2.1. Class *svm\_train*

*Class* ini bertugas menangani proses pelatihan SVReg.

1. Melakukan *cross validation* terhadap data set training dengan memanggil *method svm\_cross\_validation* dari *class svm*. Method ini membutuhkan file input berupa *svm\_problem* yang telah diubah di *class mainIntercafe*, *svm\_parameter* yang didapatkan dari input pengguna dan *nr\_fold*. *Cross validation* ini digunakan sebagai bahan evaluasi model SVM yang akan dibentuk.

```
svm.svm_cross_validation(prob,param,nr_fold,target);
```

Gambar 4.1 *Source code* memanggil *method svm\_cross\_validation*

2. Menghitung nilai MSE dari proses pelatihan yang dilakukan oleh SVReg.

```

for(i=0;i<prob.l;i++){
    double y = prob.y[i];
    double v = target[i];
    total_error += (v-y)*(v-y);
    sumv += v;
    sumy += y;
    sumvv += v*v;
    sumyy += y*y;
    sumvy += v*y;
}

mse = total_error/prob.l;
sco = ((prob.l*sumvy-sumv*sumy)*(prob.l*sumvy-
sumv*sumy))/((prob.l*sumvv-sumv*sumv)*(prob.l*sumyy-
sumy*sumy));

```

Gambar 4.2 *Source code* menghitung MSE

#### 4.2.2. Class svm\_predict

*Class* svm\_predict bertugas menangani proses prediksi SVReg.

1. Membaca file input data set prediction yang berisi data yang akan diprediksi nilainya oleh SVReg. Tiap baris data masukan akan disimpan ke dalam bentuk svm\_node. *Class* svm\_node merupakan kelas untuk menyimpan atribut dan lainnya

```

while(true)
{
    String line = input.readLine();
    if(line == null) break;

    StringTokenizer st = new
StringTokenizer(line, " \t\n\r\f:");

    double target = atof(st.nextToken());
    int m = st.countTokens()/2;
    svm_node[] x = new svm_node[m];
    for(int j=0;j<m;j++)

```

```

        {
            x[j] = new svm_node();
            x[j].index = atoi(st.nextToken());
            x[j].value = atof(st.nextToken());
        }

```

Gambar 4.3 *Source code* membaca file input dan diubah ke dalam bentuk svm\_node

2. Melakukan proses prediksi dengan memanggil *method* svm\_predict dari *class* svm. *Method* ini membutuhkan file input berupa model SVM dan data *time series* yang akan diprediksi. Hasil prediksi dituliskan ke dalam *file output*.

```

v = svm.svm_predict(model,x);
output.writeBytes(v+"\n");

```

Gambar 4.4 *Source code* memanggil *method* svm\_predict

3. Menghitung nilai MSE dan MAPE dari proses prediksi yang dilakukan oleh SVReg.

```

if(v == target)
    ++correct;
error += (v-target)*(v-target);
error2 += Math.abs(target-v)/target;

sumv += v;
sumy += target;
sumvv += v*v;
sumyy += target*target;
sumvy += v*target;
++total;

i++;
}

mse = error/total;
mape = error2/total*100;
sco = ((total*sumvy-sumv*sumy)*(total*sumvy-
sumv*sumy))/((total*sumvv-sumv*sumv)*(total*sumyy-sumy*sumy));

```

Gambar 4.5 *Source code* menghitung MSE dan MAPE

### 4.2.3. Class mainInterface

*Class* ini berisi desain antarmuka dari SVReg dan penggunaan *method* dari *class* svm\_train.java dan svm\_predict.java. Desain antarmuka seperti yang telah dijelaskan pada sub bab 3.3.4.

*Class* ini menangani dua form, yakni form proses pelatihan dan form proses prediksi.

#### 4.2.3.1. Proses pelatihan

1. Mengambil nilai parameter dari user interface.

```
getParameterValue();
```

2. Mengubah file input data set training dengan format libsvm ke dalam bentuk svm problem. Untuk atributnya disimpan di dalam svm\_node.

```
readProblem();
```

```
public void readProblem() {
    try {
        BufferedReader fp = new BufferedReader(new
        FileReader(dataSetTraining));
        Vector<Double> vy = new Vector<Double>();
        Vector<svm_node[]> vx = new Vector<svm_node[]>();
        int max_index = 0;

        while (true) {
            try {
                String line = fp.readLine();
                if (line == null) {
                    break;
                }

                StringTokenizer st = new
                StringTokenizer(line, " \t\n\r\f:");

                vy.addElement(svm_train.atof(st.nextToken()));
                int m = st.countTokens() / 2;
                svm_node[] x = new svm_node[m];
                for (int j = 0; j < m; j++) {
```

```

        x[j] = new svm_node();
        x[j].index =
svm_train.atoi(st.nextToken());
        x[j].value =
svm_train.atof(st.nextToken());
    }

    if (m > 0) {
        max_index = Math.max(max_index, x[m -
1].index);
    }
    vx.addElement(x);
} catch (IOException i) {
    System.out.println("Error on reading
problem");
}

}

prob = new svm_problem();
prob.l = vy.size();
prob.x = new svm_node[prob.l][];
for (int i = 0; i < prob.l; i++) {
    prob.x[i] = vx.elementAt(i);
}
prob.y = new double[prob.l];
for (int i = 0; i < prob.l; i++) {
    prob.y[i] = vy.elementAt(i);
}

if (param.gamma == 0) {
    param.gamma = 1.0 / max_index;
}

fp.close();
} catch (IOException i) {
}
}

```

Gambar 4.6 *Source code* mengubah file input ke dalam svm\_problem

3. Melakukan pengecekan parameter jika ada yang tidak sesuai dengan *class svm\_parameter* dan akan memberikan output apabila ada parameter yang tidak sesuai. *Method* ini harus dipanggil sebelum memanggil *method svm\_cross\_validation* dan *svm\_train*.



```

error_msg = svm.svm_check_parameter(prob, param);
if (error_msg != null) {
    textAreaModel.append("Error = " + error_msg + "\n");
}

```

Gambar 4.7 *Source code* memanggil *method* `svm_check_parameter`

4. Melakukan pengecekan nilai parameter *cross validation*, jika nilainya tidak sama dengan 0 maka akan dilakukan pelatihan dengan memanggil *method* `do_cross_validation` pada *class* `svm_train` kemudian nilai evaluasi ditampilkan pada *textAreaModel* dan jika nilainya sama dengan 0 maka *method* `svm_train` pada *class* `svm` akan dipanggil dan model akan disimpan.

```

crossValidationValue =
Integer.parseInt(textFieldCrossValidation.getText());
textAreaModel.append("EVALUATION\n-----
-----" + "-----\nSVR Type      = " +
comboBoxSvmType.getSelectedItem() + "\nKernel Type = " +
comboBoxKernelFunction.getSelectedItem() + "\n\n");

    if (crossValidationValue != 0) {
        t.do_cross_validation(prob, param, nr_fold);
        textAreaModel.append("Mean Squared Error = " +
svm_train.getMSE() +
"\nCROSS Validation Squared
correlation coefficient = " + svm_train.getSCO() + "\n");
        textAreaModel.append("nSV = " + svm.get_nSV() + "\n");

    } else {
        model = svm.svm_train(prob, param);
        try {
            svm.svm_save_model(model_file_name,
model);
        } catch (IOException i) {
        }
        textAreaModel.append(".....Model saved!");
    }
}

```

Gambar 4.8 *Source code* memanggil *method* `do_cross_validation` dan `svm_train`

#### 4.2.3.2. Proses prediksi

1. Mengambil masukan data set prediction yang berisi data berformat libsvm yang akan diprediksi dan membuat satu file hasil prediksi berformat .output yang akan digunakan untuk menyimpan hasil prediksi SVReg.

```
try {
    FileReader berkasMasukanDataSetPrediction = new
    FileReader(dataSetPrediction);
    BufferedReader input = new
    BufferedReader(berkasMasukanDataSetPrediction);
    DataOutputStream output = new DataOutputStream(new
    FileOutputStream(dataSetPrediction + ".output"));
```

Gambar 4.9 *Source code* membaca file input dan membuat file output

2. Mengambil file model SVM yang akan digunakan sebagai acuan dalam melakukan prediksi kemudian di-load. Format file ini adalah .model.

```
try {
    model = svm.svm_load_model(dataModel);
}
```

Gambar 4.10 *Source code* memanggil *method* svm\_load\_model

3. Melakukan proses prediksi dengan memanggil *method* *predict* dari class *svm\_predict* kemudian menutup file input dan file output.

```
try {
    svm_predict.predict(input, output, model,
    predict_probability, jumlah);
    input.close();
    output.close();
}
```

Gambar 4.11 *Source code* memanggil *method* svm\_predict

#### 4. Menampilkan hasil prediksi dan evaluasi nilai MSE dan MAPE.

```

textAreaPerformance.append("Mean Squared Error = " +
svm_predict.getMSE() + "\nMean Absolute Percentage Error = " +
svm_predict.getMAPE() +
"%\nCross Validation Squared Correlation
Coefficient = " + svm_predict.getSCO() + "\n");

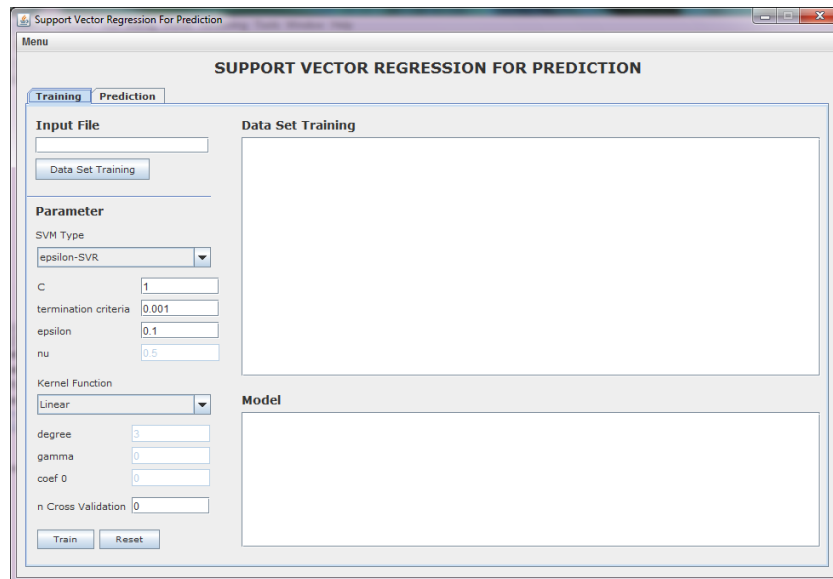
int x = svm_predict.getIndex();
double[] s = svm_predict.getArray();
for (int d = 0; d < x; d++) {
    textAreaPrediction.append((d+1) + " -- " + s[d] +
"\n");

```

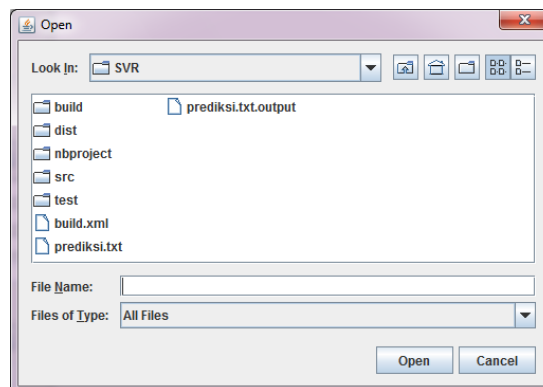
Gambar 4.12 *Source code* menampilkan nilai evaluasi dan nilai prediksi

### 4.3 Implementasi Aplikasi

Sesuai dengan rancangan desain pada sub bab 3.3.4, maka saat SVReg dibuka, form pertama yang muncul yaitu form training seperti pada gambar 4.12. Pengguna dapat membuka file input data set training untuk poses pelatihan dengan cara meng-klik tombol Data Set Training. Jika tombol ini di-klik maka akan muncul JDialog yaitu Chooser untuk memilih file yang akan digunakan sebagai file input data set training SVReg seperti pada gambar 4.13.



Gambar 4.13 Form training



Gambar 4.14 JDialog untuk memilih file data set training

Pada text area Data Set Training akan ditampilkan isi dari file input, berupa kumpulan data yang digunakan dalam proses pelatihan.

Gambar 4.15 Form training sebelum sistem dijalankan

Setelah memilih file input data set training, pengguna menentukan nilai parameter dalam proses pelatihan. Beberapa parameter yang harus diperhatikan dalam proses pelatihan, yakni SVM type, epsilon dan kernel type. Untuk SVM type, pengguna dapat memilih akan menggunakan epsilon SVR atau nu SVR. Masing-masing tipe SVM memiliki parameter yang harus diisi. Epsilon SVR memiliki parameter C dengan nilai default 1 dan epsilon dengan nilai default 0.1 sedangkan nu SVR memiliki parameter C dengan nilai default 1 dan nu dengan nilai default 0.5. Parameter *termination criteria* digunakan sebagai *stopping criteria* dalam proses pelatihan, nilai defaultnya adalah 0.001.

Untuk kernel type ada beberapa tipe kernel yang bisa digunakan untuk memetakan data, yaitu Linear, Polynomial, RBF dan Sigmoid. Kernel Linear tidak memiliki parameter yang harus diisi nilainya. Kernel Polynomial memiliki parameter degree, parameter gamma dan parameter coef0. Kernel RBF memiliki parameter

gamma. Sedangkan kernel Sigmoid memiliki parameter gamma dan coef0. Nilai default untuk parameter degree adalah 3, parameter gamma adalah 0 dan parameter coef0 adalah 0.

Untuk mendapatkan hasil evaluasi dari proses training, parameter *n-cross validation* harus diisi dengan nilai lebih besar dari 0 lalu klik tombol Train. *n-cross validation* digunakan untuk melakukan validasi terhadap proses pelatihan sehingga didapatkanlah nilai MSE dan SCO yang akan ditampilkan di *text area* evaluation. Dari nilai MSE dan SCO ini tingkat akurasi proses pelatihan dapat diketahui.

SCO merupakan kepanjangan dari Squared Correlation Coefisien yang menunjukkan seberapa baik garis regresi yang dihasilkan terhadap data poin.

Gambar 4.16 Form setelah sistem dijalankan

Gambar 4.17 Form training menyimpan model

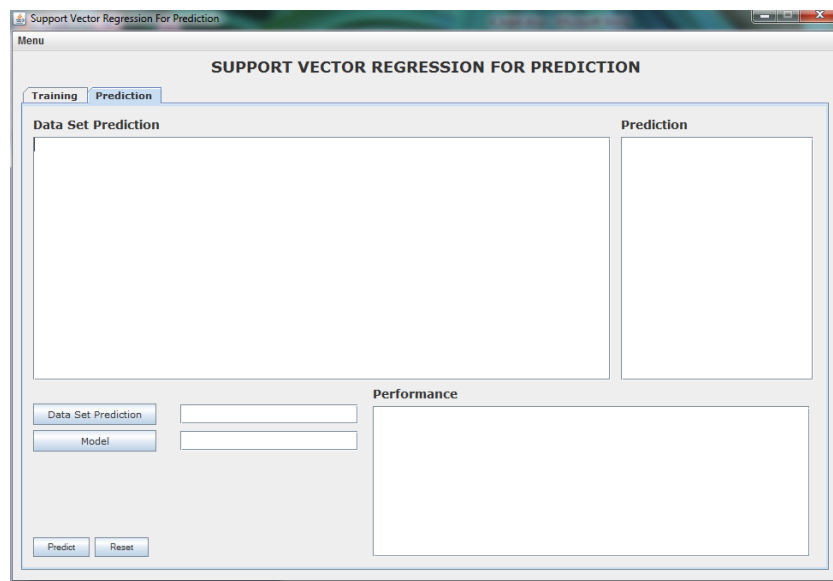
Setelah didapatkan nilai yang paling kecil, parameter *n-cross validation* diset dengan nilai 0 untuk menyimpan model arsitektur hasil pelatihan. Model arsitektur disimpan dalam bentuk file dengan ekstensi .model yang disimpan dalam direktori yang sama seperti file Data Set Training dengan nama file berikut

*FileInputDataSetTraining.model*

Model inilah yang nanti akan digunakan dalam proses prediksi. File model berisi berbagai macam data yang merupakan hasil proses pelatihan diperlukan dalam proses prediksi, meliputi SVM type, kernel type, jumlah *support vector*, rho dan list *support vector*.

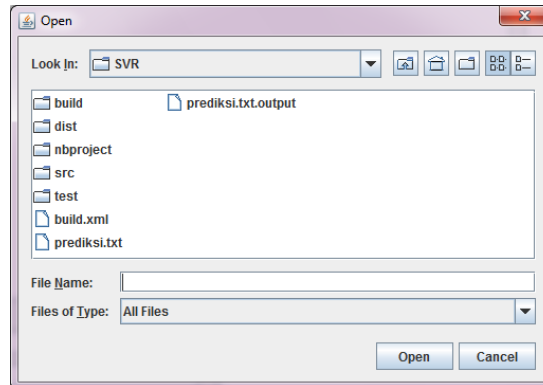
SVReg mempunyai dua form. Form pertama merupakan form training sedangkan form kedua adalah form prediction. Pada form prediction pengguna dapat melakukan proses prediksi. Pengguna dapat memilih file input data set prediction

yang akan diprediksi nilainya dengan meng-klik tombol Data Set Prediction. Jika tombol ini di-klik maka akan muncul JDialog berupa JFileChooser untuk memilih file input. Kemudian pada *text area* Data Set Prediction akan ditampilkan isi file input Data Set Prediction. Di samping menentukan file input Data Set Prediction, pengguna juga harus menentukan file model yang akan digunakan sebagai acuan dalam proses prediksi dengan meng-klik tombol Model. Sama seperti pada tombol Data Set Prediction, jika tombol Model di-klik, maka akan muncul JDialog berupa JFileChooser yang bisa digunakan oleh pengguna untuk menentukan file model.

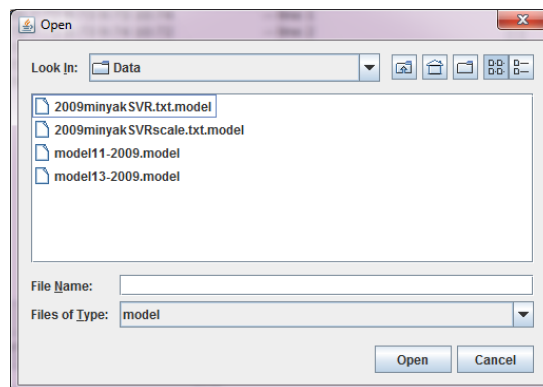


Gambar 4.18 Form prediction





Gambar 4.19 JDIALOG untuk memilih file data set prediction



Gambar 4.20 JDIALOG untuk memilih file model

| Data Set Prediction |         |         |         |         |         |         |         |         |         |          | Prediction |
|---------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|------------|
| 70.82               | 1:72.47 | 2:72.04 | 3:69.71 | 4:71.55 | 5:68.97 | 6:65.79 | 7:66.02 | 8:66.84 | 9:66.71 | 10:70.61 | -- line    |
| 69.95               | 1:72.04 | 2:69.71 | 3:71.55 | 4:68.97 | 5:65.79 | 6:66.02 | 7:66.84 | 8:66.71 | 9:70.61 | 10:70.82 | -- line    |
| 70.41               | 1:69.71 | 2:71.55 | 3:68.97 | 4:65.79 | 5:66.02 | 6:66.84 | 7:66.71 | 8:70.61 | 9:70.82 | 10:69.95 | -- line    |
| 70.88               | 1:71.55 | 2:68.97 | 3:65.79 | 4:66.02 | 5:66.84 | 6:66.71 | 7:70.61 | 8:70.82 | 9:69.95 | 10:70.41 | -- line    |
| 69.57               | 1:68.97 | 2:65.79 | 3:66.02 | 4:66.84 | 5:66.71 | 6:70.61 | 7:70.82 | 8:69.95 | 9:70.41 | 10:70.88 | -- line    |
| 71.69               | 1:65.79 | 2:66.02 | 3:66.84 | 4:66.71 | 5:70.61 | 6:70.82 | 7:69.95 | 8:70.41 | 9:70.88 | 10:69.57 | -- line    |
| 71.77               | 1:66.02 | 2:66.84 | 3:66.71 | 4:70.61 | 5:70.82 | 6:69.95 | 7:70.41 | 8:70.88 | 9:69.57 | 10:71.69 | -- line    |
| 73.27               | 1:66.84 | 2:66.71 | 3:70.61 | 4:70.82 | 5:69.95 | 6:70.41 | 7:70.88 | 8:69.57 | 9:71.69 | 10:71.77 | -- line    |
| 74.15               | 1:66.71 | 2:70.61 | 3:70.82 | 4:69.95 | 5:70.41 | 6:70.88 | 7:69.57 | 8:71.69 | 9:71.77 | 10:73.27 | -- line    |
| 75.18               | 1:70.61 | 2:70.82 | 3:69.95 | 4:70.41 | 5:70.88 | 6:69.57 | 7:71.69 | 8:71.77 | 9:73.27 | 10:74.15 | -- line    |
| 77.58               | 1:70.82 | 2:69.95 | 3:70.41 | 4:70.88 | 5:69.57 | 6:71.69 | 7:71.77 | 8:73.27 | 9:74.15 | 10:75.18 | -- line    |
| 78.53               | 1:69.95 | 2:70.41 | 3:70.88 | 4:69.57 | 5:71.69 | 6:71.77 | 7:73.27 | 8:74.15 | 9:75.18 | 10:77.58 | -- line    |
| 79.61               | 1:70.41 | 2:70.88 | 3:69.57 | 4:71.69 | 5:71.77 | 6:73.27 | 7:74.15 | 8:75.18 | 9:77.58 | 10:78.53 | -- line    |
| 79.09               | 1:70.88 | 2:69.57 | 3:71.69 | 4:71.77 | 5:73.27 | 6:74.15 | 7:75.18 | 8:77.58 | 9:78.53 | 10:79.61 | -- line    |
| 81.37               | 1:69.57 | 2:71.69 | 3:71.77 | 4:73.27 | 5:74.15 | 6:75.18 | 7:77.58 | 8:78.53 | 9:79.61 | 10:79.09 | -- line    |
| 81.19               | 1:71.69 | 2:71.77 | 3:73.27 | 4:74.15 | 5:75.18 | 6:77.58 | 7:78.53 | 8:79.61 | 9:79.09 | 10:81.37 | -- line    |
| 81.50               | 1:71.77 | 2:73.27 | 3:74.15 | 4:75.18 | 5:77.58 | 6:78.53 | 7:79.61 | 8:79.09 | 9:81.37 | 10:81.19 | -- line    |

Gambar 4.21 Form prediction sebelum sistem dijalankan

Setelah file input Data Set Prediction dan file Model ditentukan oleh pengguna, maka proses prediksi bisa segera dilakukan dengan meng-klik tombol Predict. Hasil prediksi bisa dilihat dalam *text area* Prediction dan disimpan dalam bentuk file berekstensi .output yang disimpan dalam direktori yang sama dengan file input dengan nama file berikut

*FileInputDataSetPrdiction.output*

Nilai evaluasinya yang berupa nilai MSE, MAPE dan SCO bisa dilihat di *text area* Performance.

**SUPPORT VECTOR REGRESSION FOR PREDICTION**

**Training Prediction**

**Data Set Prediction**

|       |         |         |         |         |         |         |         |         |         |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 70.82 | 1:72.47 | 2:72.04 | 3:69.71 | 4:71.55 | 5:68.97 | 6:65.79 | 7:66.02 | 8:66.84 | 9:66.71 | 10:70.61 |
| 69.95 | 1:72.04 | 2:69.71 | 3:71.55 | 4:68.97 | 5:65.79 | 6:66.02 | 7:66.84 | 8:66.71 | 9:70.61 | 10:70.82 |
| 70.41 | 1:69.71 | 2:71.55 | 3:68.97 | 4:65.79 | 5:66.02 | 6:66.84 | 7:66.71 | 8:70.61 | 9:70.82 | 10:69.95 |
| 70.88 | 1:71.55 | 2:68.97 | 3:65.79 | 4:66.02 | 5:66.84 | 6:66.71 | 7:70.61 | 8:70.82 | 9:69.95 | 10:70.41 |
| 69.57 | 1:68.97 | 2:65.79 | 3:66.02 | 4:66.84 | 5:66.71 | 6:70.61 | 7:70.82 | 8:69.95 | 9:70.41 | 10:70.88 |
| 71.69 | 1:65.79 | 2:66.02 | 3:66.84 | 4:66.71 | 5:70.61 | 6:70.82 | 7:69.95 | 8:70.41 | 9:70.88 | 10:69.57 |
| 71.77 | 1:66.02 | 2:66.84 | 3:66.71 | 4:70.61 | 5:70.82 | 6:69.95 | 7:70.41 | 8:70.88 | 9:69.57 | 10:71.69 |
| 73.27 | 1:66.84 | 2:66.71 | 3:70.61 | 4:70.82 | 5:69.95 | 6:70.41 | 7:70.88 | 8:69.57 | 9:71.69 | 10:71.77 |
| 74.15 | 1:66.71 | 2:70.61 | 3:70.82 | 4:69.95 | 5:70.41 | 6:70.88 | 7:69.57 | 8:71.69 | 9:71.77 | 10:73.27 |
| 75.18 | 1:70.61 | 2:70.82 | 3:69.95 | 4:70.41 | 5:70.88 | 6:69.57 | 7:71.69 | 8:71.77 | 9:73.27 | 10:74.15 |
| 77.58 | 1:70.82 | 2:69.95 | 3:70.41 | 4:70.88 | 5:69.57 | 6:71.69 | 7:71.77 | 8:73.27 | 9:74.15 | 10:75.18 |
| 78.53 | 1:69.95 | 2:70.41 | 3:70.88 | 4:69.57 | 5:71.69 | 6:71.77 | 7:73.27 | 8:74.15 | 9:75.18 | 10:77.58 |
| 79.61 | 1:70.41 | 2:70.88 | 3:69.57 | 4:71.69 | 5:71.77 | 6:73.27 | 7:74.15 | 8:75.18 | 9:77.58 | 10:78.53 |
| 79.09 | 1:70.88 | 2:69.57 | 3:71.69 | 4:71.77 | 5:73.27 | 6:74.15 | 7:75.18 | 8:77.58 | 9:78.53 | 10:79.61 |
| 81.37 | 1:69.57 | 2:71.69 | 3:71.77 | 4:73.27 | 5:74.15 | 6:75.18 | 7:77.58 | 8:78.53 | 9:79.61 | 10:79.09 |
| 81.19 | 1:71.69 | 2:71.77 | 3:73.27 | 4:74.15 | 5:75.18 | 6:77.58 | 7:78.53 | 8:79.61 | 9:79.09 | 10:81.37 |
| 80.50 | 1:71.77 | 2:73.27 | 3:74.15 | 4:75.18 | 5:77.58 | 6:78.53 | 7:79.61 | 8:79.09 | 9:81.37 | 10:81.19 |

**Prediction**

|                         |
|-------------------------|
| 6 -- 71.29903303900079  |
| 6 -- 69.98034613342534  |
| 7 -- 71.37943917930752  |
| 8 -- 72.09589810874823  |
| 9 -- 72.7516619919792   |
| 10 -- 74.30211035780866 |
| 11 -- 75.11855081941977 |
| 12 -- 77.662729371967   |
| 13 -- 78.14381897957574 |
| 14 -- 79.7596065430138  |
| 15 -- 78.87134526849186 |
| 16 -- 81.20085055152813 |
| 17 -- 81.13366484250801 |
| 18 -- 80.25107504155531 |
| 19 -- 79.07678454883208 |
| 20 -- 79.29362017712484 |
| 21 -- 77.9704480612003  |

**Performance**

Mean Squared Error = 1.832351033701872  
Mean Absolute Percentage Error = 1.5165353655893725%  
Cross Validation Squared Correlation Coefficient = 0.908075408948565

Data Set Prediction: Asli\dataMinyakAsliSetPrediction.txt  
Model: JilonLinearDataOilSetTraining.model

Predict Reset

Gambar 4.22 Form prediction setelah sistem dijalankan

## **4.4 Pengujian**

### **4.4.1. Data, Parameter dan Prosedur Pengujian**

Sesuai dengan rancangan pengujian sistem, untuk mengetahui kinerja SVReg dalam melakukan proses prediksi digunakanlah perhitungan MSE dan MAPE untuk mendapatkan tingkat akurasi prediksi SVReg.

Seperti yang sudah dijelaskan sebelumnya, ada dua data yang digunakan untuk menguji tingkat akurasi SVReg, yakni data harga minyak mentah dunia berdasarkan WTI dan nilai tukar Rupiah terhadap US Dollar.

Data pertama yang digunakan untuk melakukan pengujian terhadap kinerja SVReg adalah data harga minyak mentah dunia pada tahun 2009, tepatnya dari tanggal 2 Januari 2009 hingga tanggal 29 Oktober 2009. Data ini kemudian diubah ke dalam format libsvm. Data ini kemudian dipecah menjadi dua dataset. Data set pertama disebut data set training, yakni data dari 2 Januari 2009 hingga 30 September 2009 yang akan digunakan sebagai data pelatihan SVReg yang nantinya digunakan untuk membentuk model SVM. Data sisanya, yakni data dari tanggal 1 Oktober 2009 hingga 29 Oktober 2009 digunakan sebagai data set prediction yang akan diprediksi untuk mengetahui tingkat akurasi prediksi SVReg.

Data kedua yang digunakan untuk melakukan pengujian terhadap kinerja SVReg adalah data nilai tukar Rupiah terhadap Dollar Amerika pada tahun 2009, tepatnya dari tanggal 1 Januari 2009 hingga tanggal 31 Oktober 2009. Sama seperti data minyak, data nilai tukar Rupiah ini kemudian diubah ke dalam format libsvm dan dipecah menjadi dua dataset. Data mulai dari tanggal 1 Januari 2009 hingga 30

September 2009 digunakan sebagai data pelatihan dan data dari tanggal 1 Oktober hingga 31 Oktober digunakan sebagai data pengujian.

*Forecasting horizon* untuk penelitian kali ini menggunakan 10 hari dengan pertimbangan data 10 hari akan cukup memberikan informasi dalam proses pelatihan namun juga tidak terlalu panjang untuk menjaga persistensi.

Untuk parameter dalam pengujian, digunakan DTREG ( <http://dtreg.com> ) untuk mendapatkan kombinasi nilai parameter yang bisa memberikan hasil training yang optimal. Oleh karena itu ada beberapa pengaturan nilai parameter awal sebelum pengujian dilakukan. Dengan memasukkan jangkauan nilai tertentu ke dalam tool ini, bisa didapatkan nilai parameter C, epsilon, nu, degree, gamma dan coefisien.

Data harga minyak mentah dunia dan nilai tukar Rupiah yang telah diubah ke dalam format SVM dan kombinasi nilai parameter yang didapatkan selanjutnya digunakan sebagai input pada proses pelatihan SVReg. Pengujian untuk membentuk tiap model dilakukan beberapa kali terhadap file input data set training yang sama dengan parameter yang berbeda. Masing-masing pengujian akan dievaluasi dengan menggunakan cross validation. Sepuluh cross validation digunakan untuk menguji kinerja SVReg dalam melakukan prediksi.

Detail kombinasi nilai parameter dan skor MSE bisa dilihat pada tabel di bawah ini.

Tabel 4.1 Parameter untuk data harga minyak

| SVM Type    | Kernel Type | Cost C   | Epsilon  | Nu       | Degree | Gamma     | Coef 0    | MSE        | SCO      |
|-------------|-------------|----------|----------|----------|--------|-----------|-----------|------------|----------|
| Epsilon SVR | Linear      | 2.782559 | 0.00001  |          |        |           |           | 3.935765   | 0.971500 |
|             | Linear      | 7.937005 | 0.00001  |          |        |           |           | 3.982137   | 0.971181 |
|             | Linear      | 10       | 0.000001 |          |        |           |           | 4.035812   | 0.970842 |
|             | Linear      | 6.80395  | 0.00001  |          |        |           |           | 4.048580   | 0.970648 |
|             | Linear      | 10       | 0.00001  |          |        |           |           | 4.091144   | 0.970387 |
|             | Linear      | 7.937005 | 0.000029 |          |        |           |           | 4.108399   | 0.970233 |
|             | Linear      | 29.24018 | 0.00001  |          |        |           |           | 4.148927   | 0.969893 |
| Epsilon SVR | Poly        | 21.54435 | 0.00001  |          | 3      | 0.0001840 | 15.848938 | 4.227638   | 0.969556 |
|             | Poly        | 10000    | 0.000001 |          | 3      | 0.0000120 | 4.378276  | 4.312222   | 0.969018 |
|             | Poly        | 10000    | 0.000001 |          | 3      | 0.0000110 | 4.449062  | 4.393418   | 0.968433 |
|             | Poly        | 0.359381 | 0.00001  |          | 3      | 0.0010620 | 94.154829 | 4.705932   | 0.966437 |
|             | Poly        | 129.155  | 0.000001 |          | 4      | 0.0000100 | 13.648855 | 5.151611   | 0.962753 |
|             | Poly        | 35938.14 | 0.000001 |          | 3      | 0.0000120 | 4.378276  | 5.424810   | 0.961062 |
| Epsilon SVR | RBF         | 215443.5 | 0.000001 |          |        | 0.0000001 |           | 3.889628   | 0.971761 |
|             | RBF         | 215443.5 | 0.000001 |          |        | 0.0000010 |           | 4.027780   | 0.970892 |
|             | RBF         | 215443.5 | 0.000001 |          |        | 0.0000100 |           | 4.647542   | 0.966601 |
|             | RBF         | 599484.3 | 0.000001 |          |        | 0.0000100 |           | 4.736325   | 0.965849 |
|             | RBF         | 27905.03 | 0.00001  |          |        | 0.0000450 |           | 4.747283   | 0.966080 |
|             | RBF         | 2411.681 | 0.00001  |          |        | 0.0004970 |           | 7.093070   | 0.950676 |
| Epsilon SVR | Sigmoid     | 35858.2  | 0.053589 |          |        | 0.661734  | 4.641579  | 144.930352 | 0.036316 |
|             |             |          |          |          |        |           |           |            |          |
| Nu SVR      | Linear      | 0.116155 |          | 0.774626 |        |           |           | 3.889261   | 0.971814 |
|             | Linear      | 0.008315 |          | 0.9      |        |           |           | 3.942436   | 0.971524 |
|             | Linear      | 2.782559 |          | 0.9      |        |           |           | 3.994060   | 0.970997 |
|             | Linear      | 0.11615  |          | 0.85014  |        |           |           | 3.997605   | 0.970990 |
|             | Linear      | 5.995285 |          | 0.308905 |        |           |           | 4.087516   | 0.970301 |
|             | Linear      | 215.4435 |          | 0.324959 |        |           |           | 4.095237   | 0.970287 |
| Nu SVR      | Poly        | 3642.629 |          | 0.604539 | 3      | 0.000001  | 11.53824  | 4.305751   | 0.969716 |
|             | Poly        | 3642.629 |          | 0.104539 | 3      | 0.00001   | 11.53824  | 4.665670   | 0.966235 |
|             | Poly        | 3642.629 |          | 0.604539 | 3      | 0.00001   | 11.53824  | 4.803183   | 0.965906 |
| Nu SVR      | RBF         | 961.1099 |          | 0.850619 |        | 0.000006  |           | 3.977620   | 0.971161 |
|             | RBF         | 961.1099 |          | 0.850619 |        | 0.00006   |           | 4.061579   | 0.970532 |
|             | RBF         | 967.6665 |          | 0.747605 |        | 0.00006   |           | 4.257452   | 0.969176 |
|             | RBF         | 1315138  |          | 0.899493 |        | 0.000008  |           | 4.727343   | 0.965965 |
| Nu SVR      | Sigmoid     | 12.94351 |          | 0.8499   |        | 0.972271  | 0.622837  | 150.196886 | 0.064230 |

Tabel 4.2 Parameter untuk data nilai tukar Rupiah terhadap US Dollar

| SVM Type    | Kernel Type | Cost C      | Epsilon | Nu       | Degree | Gamma    | Coef 0   | MSE            | SCO      |
|-------------|-------------|-------------|---------|----------|--------|----------|----------|----------------|----------|
| Epsilon SVR | Linear      | 0.429753    | 0.00004 |          |        |          |          | 18155.1466     | 0.972017 |
|             | Linear      | 146.586659  | 0.00001 |          |        |          |          | 727136916.3550 | 0.000084 |
| Epsilon SVR | RBF         | 2022.499616 | 0.00001 |          |        | 0.000079 |          | 471713.9564    | 0.270955 |
|             | RBF         | 385013.9355 | 0.00001 |          |        | 0.000138 |          | 542333.5076    | 0.124528 |
|             | RBF         | 69198.75981 | 0.00001 |          |        | 0.000521 |          | 597463.6042    | 0.009176 |
| Epsilon SVR | Sigmoid     | 12887.05067 | 0.00001 |          |        | 0.084112 | 1.697441 | 706763.7525    | 0.025850 |
|             |             |             |         |          |        |          |          |                |          |
| Nu SVR      | Linear      | 0.466129    |         | 0.726752 |        |          |          | 12188.0428     | 0.979941 |
|             | Linear      | 1.846876    |         | 0.88322  |        |          |          | 111337.5768    | 0.822696 |
|             | Linear      | 11634.59084 |         | 0.6      |        |          |          | 2241475053749  | 0.007032 |
|             | Linear      | 33420.99636 |         | 0.877334 |        |          |          | 29372520956586 | 0.000112 |
| Nu SVR      | RBF         | 11634.8986  |         | 0.849557 |        | 0.000036 |          | 278281.9507    | 0.626361 |
|             | RBF         | 54232.22866 |         | 0.707646 |        | 0.000036 |          | 288132.9120    | 0.608710 |
|             | RBF         | 146.586659  |         | 0.6      |        | 0.000042 |          | 511440.3478    | 0.369627 |
|             | RBF         | 559920.5122 |         | 0.891103 |        | 0.000109 |          | 517910.5136    | 0.174095 |
|             | RBF         | 556037.4388 |         | 0.898621 |        | 0.000109 |          | 518126.9527    | 0.174160 |
| Nu SVR      | Sigmoid     | 28.236018   |         | 0.457582 |        | 0.079936 | 1.668098 | 605528.0481    | 0.015187 |
|             |             | 0.145188    |         | 0.69817  |        | 0.081577 | 0.735642 | 615893.3980    | 0.031982 |
|             |             | 12692.17414 |         | 0.898301 |        | 0.084188 | 1.715819 | 689673.2193    | 0.014726 |

Dari tabel di atas dapat diketahui jenis SVM, kernel type berikut nilai parameter yang optimal yang nanti bisa digunakan dalam membentuk model SVM terbaik sebagai acuan untuk melakukan prediksi.

Sesuai dengan prosedur pengujian yang telah dijelaskan pada sub bab 3.4, setelah dilakukan pelatihan dengan berbagai macam kombinasi parameter yang didapatkan maka akan didapatkan kombinasi parameter yang menghasilkan nilai MSE yang paling rendah. Kombinasi parameter terbaik ini akan digunakan untuk

membentuk model SVM yang nantinya akan digunakan sebagai model SVM dalam proses prediksi.

#### **4.4.2. Pengujian**

SVReg menggunakan metode SVM untuk mendapatkan bidang pemisah yang paling optimal. Ada dua jenis SVM yang bisa digunakan, yakni epsilon SVR dan nu SVR. Masing-masing jenis SVM bisa memanfaatkan salah satu dari empat fungsi kernel untuk melakukan pemetaan data ke dalam dimensi yang lebih tinggi. Oleh karena itu ada delapan model yang bisa dibentuk dalam pengujian kali ini. Model SVM terbaik dipilih berdasarkan nilai MSE yang paling rendah yang didapatkan dalam proses pelatihan.

Berdasarkan hasil pelatihan beberapa kombinasi parameter pada tabel 4.1, dapat diambil delapan kombinasi nilai parameter pelatihan data harga minyak yang menghasilkan nilai MSE terendah. Masing-masing satu kombinasi untuk tiap kernel yang tersedia. Untuk detail kombinasi parameternya dapat dilihat pada tabel 4.3.

Tabel 4.3 Kombinasi parameter terbaik data harga minyak

|         | E SVR<br>Linear | E SVR Poly | E SVR RBF | E SVR<br>Sigmoid | Nu SVR<br>Linear | Nu SVR<br>Poly | Nu SVR<br>RBF | Nu SVR<br>Sigmoid |
|---------|-----------------|------------|-----------|------------------|------------------|----------------|---------------|-------------------|
| C       | 2.782559        | 21.54435   | 215443.5  | 35858.2          | 0.116155         | 3642.629       | 961.1099      | 12.94351          |
| Epsilon | 0.00001         | 0.00001    | 0.000001  | 0.053589         |                  |                |               |                   |
| Nu      |                 |            |           |                  | 0.774626         | 0.604539       | 0.850619      | 0.8499            |
| Degree  |                 | 3          |           |                  |                  | 3              |               |                   |
| Gamma   |                 | 0.0001840  | 0.0000001 | 0.661734         |                  | 0.000001       | 0.000006      | 0.972271          |
| Coef0   |                 | 15.848938  |           | 4.641579         |                  | 11.53824       |               | 0.622837          |
| MSE     | 3.935765        | 4.227638   | 3.889628  | 144.930352       | 3.889261         | 4.305751       | 3.977620      | 150.196886        |

Untuk kombinasi nilai parameter terbaik data nilai tukar Rupiah berdasarkan tabel 4.2 dapat dilihat pada tabel 4.4.

Tabel 4.4 Kombinasi parameter terbaik data nilai tukar Rupiah

|         | E SVR<br>Linear | E SVR RBF   | E SVR<br>Sigmoid | Nu SVR<br>Linear | Nu SVR RBF  | Nu SVR<br>Sigmoid |
|---------|-----------------|-------------|------------------|------------------|-------------|-------------------|
| C       | 0.429753        | 2022.499616 | 12887.05067      | 0.466129         | 11634.8986  | 28.236018         |
| Epsilon | 0.00004         | 0.00001     | 0.00001          |                  |             |                   |
| Nu      |                 |             |                  | 0.726752         | 0.849557    | 0.457582          |
| Degree  |                 |             |                  |                  |             |                   |
| Gamma   |                 | 0.000079    | 0.084112         |                  | 0.000036    | 0.084188          |
| Coef0   |                 |             | 1.697441         |                  |             | 1.715819          |
| MSE     | 18155.1466      | 471713.9564 | 706763.7525      | 12188.0428       | 278281.9507 | 689673.2193       |

Langkah selanjutnya adalah membangun model SVM untuk melakukan prediksi dengan menggunakan parameter terbaik sesuai pada tabel 4.3.dan tabel 4.4. Kernel sigmoid tidak memberikan nilai MSE yang rendah maka kernel sigmoid tidak



begitu diperhatikan pada studi ini. Lin dan Lin (2004) juga tidak menyarankan penggunaan kernel sigmoid karena pemilihan parameter yang tepat untuk kernel ini sangat sulit dilakukan.

Untuk mengetahui kinerja prediksi dari tiap model SVM maka tiap model SVM ini akan digunakan untuk melakukan prediksi data harga minyak mentah mulai dari tanggal 1 Oktober 2009 hingga 29 Oktober 2009. Hasil prediksi terhadap data dapat dilihat pada tabel 4.5 untuk epsilon SVR dan tabel 4.6 untuk nu SVR.

Tabel 4.5 Hasil prediksi data harga minyak untuk epsilon linear

| Tanggal    | Data Asli | Epsilon Linear |       | Epsilon Polinomial |       | Epsilon RBF   |       | Epsilon Sigmoid |        |
|------------|-----------|----------------|-------|--------------------|-------|---------------|-------|-----------------|--------|
|            |           | Data Prediksi  | MAPE  | Data Prediksi      | MAPE  | Data Prediksi | MAPE  | Data Prediksi   | MAPE   |
| 10/1/2009  | \$70.82   | 70.40          | 0.60% | 70.53              | 0.41% | 70.15         | 0.94% | 60.28           | 14.89% |
| 10/2/2009  | \$69.95   | 71.42          | 2.10% | 71.28              | 1.90% | 71.27         | 1.88% | 60.28           | 13.83% |
| 10/5/2009  | \$70.41   | 69.95          | 0.66% | 68.71              | 2.42% | 69.96         | 0.64% | 60.28           | 14.39% |
| 10/6/2009  | \$70.88   | 71.05          | 0.24% | 70.70              | 0.25% | 70.65         | 0.33% | 60.28           | 14.96% |
| 10/7/2009  | \$69.57   | 71.30          | 2.49% | 71.34              | 2.55% | 71.06         | 2.15% | 60.28           | 13.36% |
| 10/8/2009  | \$71.69   | 69.98          | 2.38% | 69.75              | 2.71% | 69.78         | 2.66% | 60.28           | 15.92% |
| 10/9/2009  | \$71.77   | 71.38          | 0.54% | 71.71              | 0.08% | 71.18         | 0.83% | 60.28           | 16.02% |
| 10/12/2009 | \$73.27   | 72.10          | 1.60% | 72.53              | 1.01% | 71.89         | 1.88% | 60.28           | 17.74% |
| 10/13/2009 | \$74.15   | 72.75          | 1.89% | 72.14              | 2.71% | 72.68         | 1.99% | 60.28           | 18.71% |
| 10/14/2009 | \$75.18   | 74.30          | 1.17% | 73.94              | 1.65% | 73.99         | 1.58% | 60.28           | 19.83% |
| 10/15/2009 | \$77.58   | 75.12          | 3.17% | 74.15              | 4.43% | 74.95         | 3.39% | 60.28           | 22.31% |
| 10/16/2009 | \$78.53   | 77.66          | 1.10% | 76.46              | 2.64% | 77.24         | 1.65% | 60.28           | 23.25% |
| 10/19/2009 | \$79.61   | 78.14          | 1.84% | 76.89              | 3.42% | 77.97         | 2.06% | 60.28           | 24.29% |
| 10/20/2009 | \$79.09   | 79.76          | 0.85% | 77.44              | 2.09% | 79.40         | 0.39% | 60.28           | 23.79% |
| 10/21/2009 | \$81.37   | 78.87          | 3.07% | 76.69              | 5.75% | 78.60         | 3.41% | 60.28           | 25.92% |
| 10/22/2009 | \$81.19   | 81.20          | 0.01% | 78.40              | 3.43% | 80.72         | 0.57% | 60.28           | 25.76% |
| 10/23/2009 | \$80.50   | 81.13          | 0.79% | 78.53              | 2.44% | 80.69         | 0.23% | 60.28           | 25.12% |
| 10/26/2009 | \$78.68   | 80.25          | 2.00% | 76.22              | 3.12% | 79.93         | 1.59% | 60.28           | 23.39% |
| 10/27/2009 | \$79.55   | 79.08          | 0.59% | 75.58              | 4.99% | 78.51         | 1.30% | 60.28           | 24.23% |
| 10/28/2009 | \$77.46   | 79.29          | 2.37% | 75.55              | 2.46% | 78.76         | 1.68% | 60.28           | 22.19% |
| 10/29/2009 | \$79.87   | 77.97          | 2.38% | 74.25              | 7.03% | 77.36         | 3.15% | 60.28           | 24.53% |
|            |           |                | 1.52% |                    | 2.74% |               | 1.63% |                 | 20.21% |

Tabel 4.6 Hasil prediksi data harga minyak untuk nu linear

|            |           | nu Linear     |       | nu Polinomial |       | nu RBF        |       | nu Sigmoid    |        |
|------------|-----------|---------------|-------|---------------|-------|---------------|-------|---------------|--------|
| Tanggal    | Data Asli | Data Prediksi | MAPE  | Data Prediksi | MAPE  | Data Prediksi | MAPE  | Data Prediksi | MAPE   |
| 10/1/2009  | \$70.82   | 70.21         | 0.86% | 70.28         | 0.76% | 69.59         | 1.74% | 60.19         | 15.00% |
| 10/2/2009  | \$69.95   | 71.11         | 1.65% | 70.80         | 1.22% | 70.66         | 1.01% | 60.19         | 13.95% |
| 10/5/2009  | \$70.41   | 70.11         | 0.43% | 69.93         | 0.68% | 70.05         | 0.52% | 60.19         | 14.51% |
| 10/6/2009  | \$70.88   | 70.68         | 0.28% | 71.04         | 0.22% | 70.54         | 0.48% | 60.19         | 15.07% |
| 10/7/2009  | \$69.57   | 71.27         | 2.45% | 71.09         | 2.19% | 71.15         | 2.27% | 60.19         | 13.48% |
| 10/8/2009  | \$71.69   | 69.73         | 2.73% | 69.30         | 3.33% | 70.07         | 2.26% | 60.19         | 16.03% |
| 10/9/2009  | \$71.77   | 71.41         | 0.50% | 70.74         | 1.43% | 71.19         | 0.81% | 60.19         | 16.13% |
| 10/12/2009 | \$73.27   | 71.88         | 1.89% | 70.84         | 3.32% | 71.66         | 2.20% | 60.19         | 17.84% |
| 10/13/2009 | \$74.15   | 72.97         | 1.59% | 71.87         | 3.08% | 72.62         | 2.06% | 60.19         | 18.82% |
| 10/14/2009 | \$75.18   | 74.03         | 1.52% | 73.50         | 2.24% | 73.61         | 2.09% | 60.19         | 19.93% |
| 10/15/2009 | \$77.58   | 75.25         | 3.01% | 74.54         | 3.91% | 74.64         | 3.78% | 60.19         | 22.41% |
| 10/16/2009 | \$78.53   | 77.37         | 1.48% | 76.44         | 2.66% | 76.65         | 2.40% | 60.19         | 23.35% |
| 10/19/2009 | \$79.61   | 78.34         | 1.59% | 77.38         | 2.81% | 77.62         | 2.51% | 60.19         | 24.39% |
| 10/20/2009 | \$79.09   | 79.58         | 0.62% | 78.53         | 0.71% | 78.71         | 0.48% | 60.19         | 23.89% |
| 10/21/2009 | \$81.37   | 79.00         | 2.91% | 77.59         | 4.64% | 78.28         | 3.79% | 60.19         | 26.02% |
| 10/22/2009 | \$81.19   | 81.22         | 0.04% | 79.86         | 1.64% | 80.02         | 1.44% | 60.19         | 25.86% |
| 10/23/2009 | \$80.50   | 81.04         | 0.67% | 79.57         | 1.15% | 80.07         | 0.53% | 60.19         | 25.22% |
| 10/26/2009 | \$78.68   | 80.43         | 2.23% | 79.15         | 0.59% | 79.39         | 0.90% | 60.19         | 23.49% |
| 10/27/2009 | \$79.55   | 78.82         | 0.92% | 77.50         | 2.58% | 77.88         | 2.10% | 60.19         | 24.33% |
| 10/28/2009 | \$77.46   | 79.47         | 2.60% | 78.06         | 0.77% | 78.16         | 0.90% | 60.19         | 22.29% |
| 10/29/2009 | \$79.87   | 77.60         | 2.84% | 76.73         | 3.93% | 76.74         | 3.92% | 60.19         | 24.63% |
|            |           |               | 1.56% |               | 2.09% |               | 1.82% |               | 20.32% |

Tabel 4.7 MSE dan MAPE prediksi data harga minyak

|      | Epsilon Linear | Epsilon Polynomial | Epsilon RBF | Epsilon Sigmoid | Nu Linear | Nu Polynomial | Nu RBF | Nu Sigmoid |
|------|----------------|--------------------|-------------|-----------------|-----------|---------------|--------|------------|
| MSE  | 1.83           | 6.41               | 2.1         | 256.86          | 1.94      | 3.56          | 2.67   | 259.34     |
| MAPE | 1.52%          | 2.74%              | 1.63%       | 20.21%          | 1.56%     | 2.09%         | 1.82%  | 20.32%     |

Dari tabel 4.5, tabel 4.6 dan tabel 4.7 dapat diketahui bahwa untuk data harga minyak, kernel linear memberikan tingkat prediksi yang paling baik kemudian disusul oleh kernel RBF dan kernel Polinomial dengan hasil yang cukup berimbang antara epsilon SVR dan nu SVR.

Pada Epsilon SVR, kernel linear menghasilkan nilai MSE dan MAPE terendah, yakni 1.83 untuk MSE dan 1.52% untuk MAPE disusul oleh kernel RBF yang memberikan nilai MSE sebesar 2.1 dan nilai MAPE sebesar 1.63% kemudian disusul kernel Polinomial yang memberikan nilai MSE sebesar 6.41 dan nilai MAPE sebesar 2.74%

Seperti halnya pada Epsilon SVR, nu SVR juga memberikan hasil yang sama. Kernel Linear memberikan nilai MSE dan MAPE yang paling kecil, yakni 1.94 dan 1.56% kemudian disusul kernel RBF dengan nilai MSE sebesar 2.67 dan nilai MAPE 1.82% dan disusul kernel Polinomial dengan nilai MSE sebesar 3.56 dan nilai MAPE sebesar 2.09%.

Kernel sigmoid tidak memberikan tingkat prediksi yang baik sama sekali seperti sudah dijelaskan di awal bahwa kernel ini memang tidak disarankan penggunaannya.

Untuk mengetahui kinerja SVReg dalam melakukan prediksi terhadap data nilai tukar Rupiah terhadap Dollar Amerika dapat dilihat dalam tabel 4.8 untuk epsilon SVR dan tabel 4.9 untuk nu SVR.

Tabel 4.8 Hasil prediksi data nilai tukar Rupiah untuk epsilon linear

| Tanggal    | Data Asli | epsilon linear |          |       | epsilon RBF |         |        | epsilon sigmoid |            |        |
|------------|-----------|----------------|----------|-------|-------------|---------|--------|-----------------|------------|--------|
|            |           | Data Asli      | Error    | MAPE  | Data Asli   | Error   | MAPE   | Data Asli       | Error      | MAPE   |
| 10/1/2009  | 9661.84   | 9676.07        | 14.23    | 0.15% | 9997.58     | 335.74  | 3.47%  | 10405.80        | 743.96001  | 7.70%  |
| 10/2/2009  | 9699.32   | 9667.07        | (32.25)  | 0.33% | 10222.36    | 523.04  | 5.39%  | 10405.80        | 706.48001  | 7.28%  |
| 10/3/2009  | 9699.32   | 9620.14        | (79.18)  | 0.82% | 10150.65    | 451.33  | 4.65%  | 10405.80        | 706.48001  | 7.28%  |
| 10/4/2009  | 9643.2    | 9693.66        | 50.46    | 0.52% | 10128.73    | 485.53  | 5.03%  | 10405.80        | 762.60001  | 7.91%  |
| 10/5/2009  | 9643.2    | 9614.39        | (28.81)  | 0.30% | 10243.92    | 600.72  | 6.23%  | 10405.80        | 762.60001  | 7.91%  |
| 10/6/2009  | 9615.38   | 9600.47        | (14.91)  | 0.16% | 10272.52    | 657.14  | 6.83%  | 10405.80        | 790.42001  | 8.22%  |
| 10/7/2009  | 9551.1    | 9539.81        | (11.29)  | 0.12% | 10475.31    | 924.21  | 9.68%  | 10405.80        | 854.70001  | 8.95%  |
| 10/8/2009  | 9532.89   | 9593.83        | 60.94    | 0.64% | 10760.82    | 1227.93 | 12.88% | 10405.80        | 872.91001  | 9.16%  |
| 10/9/2009  | 9398.5    | 9458.81        | 60.31    | 0.64% | 10818.85    | 1420.35 | 15.11% | 10405.80        | 1007.30001 | 10.72% |
| 10/10/2009 | 9425.07   | 9362.65        | (62.42)  | 0.66% | 10829.22    | 1404.15 | 14.90% | 10405.80        | 980.73001  | 10.41% |
| 10/11/2009 | 9425.07   | 9393.57        | (31.50)  | 0.33% | 10829.24    | 1404.17 | 14.90% | 10405.80        | 980.73001  | 10.41% |
| 10/12/2009 | 9425.07   | 9393.78        | (31.29)  | 0.33% | 10829.24    | 1404.17 | 14.90% | 10405.80        | 980.73001  | 10.41% |
| 10/13/2009 | 9398.5    | 9321.57        | (76.93)  | 0.82% | 10829.24    | 1430.74 | 15.22% | 10405.80        | 1007.30001 | 10.72% |
| 10/14/2009 | 9363.3    | 9395.19        | 31.89    | 0.34% | 10829.24    | 1465.94 | 15.66% | 10405.80        | 1042.50001 | 11.13% |
| 10/15/2009 | 9345.79   | 9226.39        | (119.40) | 1.28% | 10829.24    | 1483.45 | 15.87% | 10405.80        | 1060.01001 | 11.34% |
| 10/16/2009 | 9363.3    | 9391.49        | 28.19    | 0.30% | 10829.24    | 1465.94 | 15.66% | 10405.80        | 1042.50001 | 11.13% |
| 10/17/2009 | 9372.07   | 9262.08        | (109.99) | 1.17% | 10829.24    | 1457.17 | 15.55% | 10405.80        | 1033.73001 | 11.03% |
| 10/18/2009 | 9398.5    | 9318.14        | (80.36)  | 0.85% | 10829.24    | 1430.74 | 15.22% | 10405.80        | 1007.30001 | 10.72% |
| 10/19/2009 | 9398.5    | 9325.25        | (73.25)  | 0.78% | 10829.24    | 1430.74 | 15.22% | 10405.80        | 1007.30001 | 10.72% |
| 10/20/2009 | 9389.67   | 9348.22        | (41.45)  | 0.44% | 10829.24    | 1439.57 | 15.33% | 10405.80        | 1016.13001 | 10.82% |
| 10/21/2009 | 9389.67   | 9337.57        | (52.10)  | 0.55% | 10829.24    | 1439.57 | 15.33% | 10405.80        | 1016.13001 | 10.82% |
| 10/22/2009 | 9354.54   | 9352.49        | (2.05)   | 0.02% | 10829.24    | 1474.70 | 15.76% | 10405.80        | 1051.26001 | 11.24% |
| 10/23/2009 | 9372.07   | 9299.01        | (73.06)  | 0.78% | 10829.24    | 1457.17 | 15.55% | 10405.80        | 1033.73001 | 11.03% |
| 10/24/2009 | 9425.07   | 9326.23        | (98.84)  | 1.05% | 10829.24    | 1404.17 | 14.90% | 10405.80        | 980.73001  | 10.41% |
| 10/25/2009 | 9433.96   | 9364.87        | (69.09)  | 0.73% | 10829.24    | 1395.28 | 14.79% | 10405.80        | 971.84001  | 10.30% |
| 10/26/2009 | 9433.96   | 9379.60        | (54.36)  | 0.58% | 10829.24    | 1395.28 | 14.79% | 10405.80        | 971.84001  | 10.30% |
| 10/27/2009 | 9451.8    | 9384.46        | (67.34)  | 0.71% | 10829.24    | 1377.44 | 14.57% | 10405.80        | 954.00001  | 10.09% |
| 10/28/2009 | 9532.89   | 9380.05        | (152.84) | 1.60% | 10829.24    | 1296.35 | 13.60% | 10405.80        | 872.91001  | 9.16%  |
| 10/29/2009 | 9596.93   | 9497.34        | (99.59)  | 1.04% | 10829.24    | 1232.31 | 12.84% | 10405.80        | 808.87001  | 8.43%  |
| 10/30/2009 | 9624.64   | 9553.00        | (71.64)  | 0.74% | 10829.24    | 1204.60 | 12.52% | 10405.80        | 781.16001  | 8.12%  |
| 10/31/2009 | 9606.15   | 9534.29        | (71.86)  | 0.75% | 10829.24    | 1223.09 | 12.73% | 10405.80        | 799.65001  | 8.32%  |
|            |           |                |          | 0.63% |             |         | 12.75% |                 |            | 9.75%  |

Tabel 4.9 Hasil prediksi data nilai tukar Rupiah untuk nu linear

|            |           | nu linear |          |       | nu RBF    |         |        | nu sigmoid |            |        |
|------------|-----------|-----------|----------|-------|-----------|---------|--------|------------|------------|--------|
| Tanggal    | Data Asli | Data asli | Error    | MAPE  | Data Asli | Error   | MAPE   | Data Asli  | Error      | MAPE   |
| 10/1/2009  | 9661.84   | 9710.12   | 48.28    | 0.50% | 9841.42   | 179.58  | 1.86%  | 10790.55   | 1128.71000 | 11.68% |
| 10/2/2009  | 9699.32   | 9613.91   | (85.41)  | 0.88% | 9882.18   | 182.86  | 1.89%  | 10790.55   | 1091.23000 | 11.25% |
| 10/3/2009  | 9699.32   | 9716.34   | 17.02    | 0.18% | 9841.53   | 142.21  | 1.47%  | 10790.55   | 1091.23000 | 11.25% |
| 10/4/2009  | 9643.2    | 9668.76   | 25.56    | 0.27% | 9794.04   | 150.84  | 1.56%  | 10790.55   | 1147.35000 | 11.90% |
| 10/5/2009  | 9643.2    | 9636.79   | (6.41)   | 0.07% | 9858.15   | 214.95  | 2.23%  | 10790.55   | 1147.35000 | 11.90% |
| 10/6/2009  | 9615.38   | 9657.50   | 42.12    | 0.44% | 9881.65   | 266.27  | 2.77%  | 10790.55   | 1175.17000 | 12.22% |
| 10/7/2009  | 9551.1    | 9560.18   | 9.08     | 0.10% | 10078.76  | 527.66  | 5.52%  | 10790.55   | 1239.45000 | 12.98% |
| 10/8/2009  | 9532.89   | 9562.96   | 30.07    | 0.32% | 10501.99  | 969.10  | 10.17% | 10790.55   | 1257.66000 | 13.19% |
| 10/9/2009  | 9398.5    | 9575.16   | 176.66   | 1.88% | 10705.76  | 1307.26 | 13.91% | 10790.55   | 1392.05000 | 14.81% |
| 10/10/2009 | 9425.07   | 9304.23   | (120.84) | 1.28% | 10885.65  | 1460.58 | 15.50% | 10790.55   | 1365.48000 | 14.49% |
| 10/11/2009 | 9425.07   | 9482.82   | 57.75    | 0.61% | 10895.15  | 1470.08 | 15.60% | 10790.55   | 1365.48000 | 14.49% |
| 10/12/2009 | 9425.07   | 9454.72   | 29.65    | 0.31% | 10896.00  | 1470.93 | 15.61% | 10790.55   | 1365.48000 | 14.49% |
| 10/13/2009 | 9398.5    | 9371.37   | (27.13)  | 0.29% | 10896.07  | 1497.57 | 15.93% | 10790.55   | 1392.05000 | 14.81% |
| 10/14/2009 | 9363.3    | 9401.15   | 37.85    | 0.40% | 10896.08  | 1532.78 | 16.37% | 10790.55   | 1427.25000 | 15.24% |
| 10/15/2009 | 9345.79   | 9298.10   | (47.69)  | 0.51% | 10896.08  | 1550.29 | 16.59% | 10790.55   | 1444.76000 | 15.46% |
| 10/16/2009 | 9363.3    | 9332.13   | (31.17)  | 0.33% | 10896.08  | 1532.78 | 16.37% | 10790.55   | 1427.25000 | 15.24% |
| 10/17/2009 | 9372.07   | 9392.21   | 20.14    | 0.21% | 10896.08  | 1524.01 | 16.26% | 10790.55   | 1418.48000 | 15.14% |
| 10/18/2009 | 9398.5    | 9295.26   | (103.24) | 1.10% | 10896.08  | 1497.58 | 15.93% | 10790.55   | 1392.05000 | 14.81% |
| 10/19/2009 | 9398.5    | 9390.17   | (8.33)   | 0.09% | 10896.08  | 1497.58 | 15.93% | 10790.55   | 1392.05000 | 14.81% |
| 10/20/2009 | 9389.67   | 9362.06   | (27.61)  | 0.29% | 10896.08  | 1506.41 | 16.04% | 10790.55   | 1400.88000 | 14.92% |
| 10/21/2009 | 9389.67   | 9361.78   | (27.89)  | 0.30% | 10896.08  | 1506.41 | 16.04% | 10790.55   | 1400.88000 | 14.92% |
| 10/22/2009 | 9354.54   | 9373.85   | 19.31    | 0.21% | 10896.08  | 1541.54 | 16.48% | 10790.55   | 1436.01000 | 15.35% |
| 10/23/2009 | 9372.07   | 9315.32   | (56.75)  | 0.61% | 10896.08  | 1524.01 | 16.26% | 10790.55   | 1418.48000 | 15.14% |
| 10/24/2009 | 9425.07   | 9362.02   | (63.05)  | 0.67% | 10896.08  | 1471.01 | 15.61% | 10790.55   | 1365.48000 | 14.49% |
| 10/25/2009 | 9433.96   | 9428.95   | (5.01)   | 0.05% | 10896.08  | 1462.12 | 15.50% | 10790.55   | 1356.59000 | 14.38% |
| 10/26/2009 | 9433.96   | 9397.38   | (36.58)  | 0.39% | 10896.08  | 1462.12 | 15.50% | 10790.55   | 1356.59000 | 14.38% |
| 10/27/2009 | 9451.8    | 9417.02   | (34.78)  | 0.37% | 10896.08  | 1444.28 | 15.28% | 10790.55   | 1338.75000 | 14.16% |
| 10/28/2009 | 9532.89   | 9417.51   | (115.38) | 1.21% | 10896.08  | 1363.19 | 14.30% | 10790.55   | 1257.66000 | 13.19% |
| 10/29/2009 | 9596.93   | 9524.72   | (72.21)  | 0.75% | 10896.08  | 1299.15 | 13.54% | 10790.55   | 1193.62000 | 12.44% |
| 10/30/2009 | 9624.64   | 9597.27   | (27.37)  | 0.28% | 10896.08  | 1271.44 | 13.21% | 10790.55   | 1165.91000 | 12.11% |
| 10/31/2009 | 9606.15   | 9556.15   | (50.00)  | 0.52% | 10896.08  | 1289.93 | 13.43% | 10790.55   | 1184.40000 | 12.33% |
|            |           |           |          | 0.50% |           |         | 12.34% |            |            | 13.81% |

Tabel 4.10 MSE dan MAPE prediksi nilai tukar Rupiah

|      | Epsilon Linear | Epsilon RBF | Epsilon Sigmoid | Nu Linear | Nu RBF     | Nu Sigmoid |
|------|----------------|-------------|-----------------|-----------|------------|------------|
| MSE  | 4701.49        | 1581519.77  | 864760.69       | 3658.165  | 1624283.87 | 1722930.78 |
| MAPE | 0.633%         | 12.767%     | 9.75%           | 0.50%     | 12.34%     | 13.81%     |

Dari tabel 4.8, tabel 4.9 dan tabel 4.10 dapat diketahui bahwa untuk data nilai tukar Rupiah, kernel linear memberikan tingkat prediksi yang paling baik kemudian disusul oleh kernel RBF.

Pada Epsilon SVR, kernel Linear menghasilkan nilai MSE dan MAPE terendah, yakni 4701.49 untuk MSE dan 1.63% untuk MAPE disusul oleh kernel Sigmoid yang memberikan nilai MSE sebesar 864760.69 dan nilai MAPE sebesar 9.75% kemudian disusul kernel RBF yang memberikan nilai MSE sebesar 1581519.77 dan nilai MAPE sebesar 12.76%

Pada nu SVR kernel Linear memberikan nilai MSE dan MAPE yang paling kecil, yakni 3658.165 dan 0.50% kemudian disusul kernel RBF dengan nilai MSE sebesar 1624283.87 dan nilai MAPE 12.34% dan disusul kernel Sigmoid dengan nilai MSE sebesar 1722930.78 dan nilai MAPE sebesar 13.81%.

Perbedaan antara epsilon SVR dan nu SVR dimungkinkan terjadi karena adanya noise pada data yang cukup besar. Oleh karena itu untuk data dengan *range* yang cukup besar perlu dilakukan normalisasi data.

Dari hasil pengujian, secara umum epsilon SVR dan nu SVR memberikan nilai yang hampir seimbang untuk kernel yang sama. Sedangkan untuk tipe kernelnya, kernel linear memberikan tingkat akurasi yang paling baik, disusul kernel RBF, kernel Polinomial dan kernel Sigmoid.

#### **4.4.3 Analisis Pengujian**

##### **4.4.3.1 Pengujian dengan Data Asli dan Data Skala**

Pengujian ini menggunakan data asli dan data skala. Untuk data skala, nilainya diskala sedemikian rupa hingga range datanya tidak terlalu besar kemudian dilakukan pelatihan kembali untuk mendapatkan model yang terbaik.

Skala pada data dilakukan karena dalam SVM, proses komputasinya berpusat pada *dot product* sehingga *feature* yang memiliki *range* besar berpotensi mendominasi hasil perhitungan akhir. Proses skala yang dilakukan pada data pelatihan harus dilakukan juga dengan cara yang sama terhadap data pelatihan.

Untuk data harga minyak proses skala data dilakukan dengan membagi data harga minyak dengan 100 untuk mendapatkan nilai yang relatif kecil. Pengujian dilakukan pada epsilon SVR dengan kernel RBF.

Tabel 4.11 Perbandingan prediksi data asli dan data skala untuk harga minyak

|              |           | Data asli     |          |       | Data skala     |               |          |       |
|--------------|-----------|---------------|----------|-------|----------------|---------------|----------|-------|
| Harga Minyak |           | Epsilon RBF   |          |       | Epsilon RBF    |               |          |       |
| Tanggal      | Data Asli | Data Prediksi | Error    | MAPE  | Data Skala     | Data Prediksi | Error    | MAPE  |
| 10/1/2009    | \$70.82   | 70.15377952   | (\$0.67) | 0.94% | 0.687333956818 | 68.73339568   | (\$2.09) | 2.95% |
| 10/2/2009    | \$69.95   | 71.26851744   | \$1.32   | 1.88% | 0.691646547794 | 69.16465478   | (\$0.79) | 1.12% |
| 10/5/2009    | \$70.41   | 69.95935592   | (\$0.45) | 0.64% | 0.693214939826 | 69.32149398   | (\$1.09) | 1.55% |
| 10/6/2009    | \$70.88   | 70.64902924   | (\$0.23) | 0.33% | 0.696223062716 | 69.62230627   | (\$1.26) | 1.77% |
| 10/7/2009    | \$69.57   | 71.06306414   | \$1.49   | 2.15% | 0.698960377756 | 69.89603778   | \$0.33   | 0.47% |
| 10/8/2009    | \$71.69   | 69.78356662   | (\$1.91) | 2.66% | 0.698392123992 | 69.8392124    | (\$1.85) | 2.58% |
| 10/9/2009    | \$71.77   | 71.17767583   | (\$0.59) | 0.83% | 0.703449033061 | 70.34490331   | (\$1.43) | 1.99% |
| 10/12/2009   | \$73.27   | 71.89008761   | (\$1.38) | 1.88% | 0.707497980617 | 70.74979806   | (\$2.52) | 3.44% |
| 10/13/2009   | \$74.15   | 72.67795054   | (\$1.47) | 1.99% | 0.714142203337 | 71.41422033   | (\$2.74) | 3.69% |
| 10/14/2009   | \$75.18   | 73.98923647   | (\$1.19) | 1.58% | 0.721561164051 | 72.15611641   | (\$3.02) | 4.02% |
| 10/15/2009   | \$77.58   | 74.95345529   | (\$2.63) | 3.39% | 0.728605180345 | 72.86051803   | (\$4.72) | 6.08% |
| 10/16/2009   | \$78.53   | 77.23802212   | (\$1.29) | 1.65% | 0.739281579566 | 73.92815796   | (\$4.60) | 5.86% |
| 10/19/2009   | \$79.61   | 77.96608349   | (\$1.64) | 2.06% | 0.749994819728 | 74.99948197   | (\$4.61) | 5.79% |
| 10/20/2009   | \$79.09   | 79.39836376   | \$0.31   | 0.39% | 0.760480095195 | 76.04800952   | (\$3.04) | 3.85% |
| 10/21/2009   | \$81.37   | 78.59698871   | (\$2.77) | 3.41% | 0.767077737076 | 76.70777371   | (\$4.66) | 5.73% |
| 10/22/2009   | \$81.19   | 80.72357425   | (\$0.47) | 0.57% | 0.778323341307 | 77.83233413   | (\$3.36) | 4.14% |
| 10/23/2009   | \$80.50   | 80.68627645   | \$0.19   | 0.23% | 0.785949565827 | 78.59495658   | (\$1.91) | 2.37% |
| 10/26/2009   | \$78.68   | 79.93035027   | \$1.25   | 1.59% | 0.790452392532 | 79.04523925   | \$0.37   | 0.46% |
| 10/27/2009   | \$79.55   | 78.51282414   | (\$1.04) | 1.30% | 0.789278750111 | 78.92787501   | (\$0.62) | 0.78% |
| 10/28/2009   | \$77.46   | 78.7609601    | \$1.30   | 1.68% | 0.790777210393 | 79.07772104   | \$1.62   | 2.09% |
| 10/29/2009   | \$79.87   | 77.35751117   | (\$2.51) | 3.15% | 0.787294456069 | 78.72944561   | (\$1.14) | 1.43% |
|              |           |               |          | 1.63% |                |               |          | 2.96% |

Pada tabel 4.11 dapat dilihat bahwa untuk data harga minyak yang *range*-nya tidak terlalu besar, skala data tidak terlalu berpengaruh terhadap prediksi. Prediksi dengan data asli dan data skala memiliki perbedaan yang cukup kecil dalam nilai MAPE, yakni sebesar 1 %.

Untuk data nilai tukar Rupiah, normalisasi dilakukan dengan membagi data harga minyak dengan 10000. Pengujian dilakukan pada epsilon SVR dengan kernel RBF.



Tabel 4.12 Perbandingan prediksi data asli dan data skala untuk data nilai tukar Rupiah

|                    |           |            | Data Asli   |         |        | Data Skala  |           |          |       |
|--------------------|-----------|------------|-------------|---------|--------|-------------|-----------|----------|-------|
| Nilai Tukar Rupiah |           |            | epsilon rbf |         |        | E RBF       |           |          |       |
| Tanggal            | Data Asli | Data Skala | Data Asli   | Error   | MAPE   | Data Skala  | Data Asli | Error    | MAPE  |
| 10/1/2009          | 9661.84   | 0.9661840  | 9997.58     | 335.74  | 3.47%  | 0.972067857 | 9720.68   | 58.84    | 0.61% |
| 10/2/2009          | 9699.32   | 0.9699320  | 10222.36    | 523.04  | 5.39%  | 0.968945325 | 9689.45   | (9.87)   | 0.10% |
| 10/3/2009          | 9699.32   | 0.9699320  | 10150.65    | 451.33  | 4.65%  | 0.969499481 | 9694.99   | (4.33)   | 0.04% |
| 10/4/2009          | 9643.2    | 0.9643200  | 10128.73    | 485.53  | 5.03%  | 0.969879957 | 9698.80   | 55.60    | 0.58% |
| 10/5/2009          | 9643.2    | 0.9643200  | 10243.92    | 600.72  | 6.23%  | 0.967287138 | 9672.87   | 29.67    | 0.31% |
| 10/6/2009          | 9615.38   | 0.9615380  | 10272.52    | 657.14  | 6.83%  | 0.965867769 | 9658.68   | 43.30    | 0.45% |
| 10/7/2009          | 9551.1    | 0.9551100  | 10475.31    | 924.21  | 9.68%  | 0.963840686 | 9638.41   | 87.31    | 0.91% |
| 10/8/2009          | 9532.89   | 0.9532890  | 10760.82    | 1227.93 | 12.88% | 0.959864574 | 9598.65   | 65.76    | 0.69% |
| 10/9/2009          | 9398.5    | 0.9398500  | 10818.85    | 1420.35 | 15.11% | 0.956799843 | 9568.00   | 169.50   | 1.80% |
| 10/10/2009         | 9425.07   | 0.9425070  | 10829.22    | 1404.15 | 14.90% | 0.948717512 | 9487.18   | 62.11    | 0.66% |
| 10/11/2009         | 9425.07   | 0.9425070  | 10829.24    | 1404.17 | 14.90% | 0.946483076 | 9464.83   | 39.76    | 0.42% |
| 10/12/2009         | 9425.07   | 0.9425070  | 10829.24    | 1404.17 | 14.90% | 0.944829801 | 9448.30   | 23.23    | 0.25% |
| 10/13/2009         | 9398.5    | 0.9398500  | 10829.24    | 1430.74 | 15.22% | 0.943436558 | 9434.37   | 35.87    | 0.38% |
| 10/14/2009         | 9363.3    | 0.9363300  | 10829.24    | 1465.94 | 15.66% | 0.941447953 | 9414.48   | 51.18    | 0.55% |
| 10/15/2009         | 9345.79   | 0.9345790  | 10829.24    | 1483.45 | 15.87% | 0.938530308 | 9385.30   | 39.51    | 0.42% |
| 10/16/2009         | 9363.3    | 0.9363300  | 10829.24    | 1465.94 | 15.66% | 0.93673373  | 9367.34   | 4.04     | 0.04% |
| 10/17/2009         | 9372.07   | 0.9372070  | 10829.24    | 1457.17 | 15.55% | 0.93672092  | 9367.21   | (4.86)   | 0.05% |
| 10/18/2009         | 9398.5    | 0.9398500  | 10829.24    | 1430.74 | 15.22% | 0.936876443 | 9368.76   | (29.74)  | 0.32% |
| 10/19/2009         | 9398.5    | 0.9398500  | 10829.24    | 1430.74 | 15.22% | 0.938423382 | 9384.23   | (14.27)  | 0.15% |
| 10/20/2009         | 9389.67   | 0.9389670  | 10829.24    | 1439.57 | 15.33% | 0.938791237 | 9387.91   | (1.76)   | 0.02% |
| 10/21/2009         | 9389.67   | 0.9389670  | 10829.24    | 1439.57 | 15.33% | 0.938655985 | 9386.56   | (3.11)   | 0.03% |
| 10/22/2009         | 9354.54   | 0.9354540  | 10829.24    | 1474.70 | 15.76% | 0.938771213 | 9387.71   | 33.17    | 0.35% |
| 10/23/2009         | 9372.07   | 0.9372070  | 10829.24    | 1457.17 | 15.55% | 0.937287107 | 9372.87   | 0.80     | 0.01% |
| 10/24/2009         | 9425.07   | 0.9425070  | 10829.24    | 1404.17 | 14.90% | 0.937584194 | 9375.84   | (49.23)  | 0.52% |
| 10/25/2009         | 9433.96   | 0.9433960  | 10829.24    | 1395.28 | 14.79% | 0.940209994 | 9402.10   | (31.86)  | 0.34% |
| 10/26/2009         | 9433.96   | 0.9433960  | 10829.24    | 1395.28 | 14.79% | 0.941573749 | 9415.74   | (18.22)  | 0.19% |
| 10/27/2009         | 9451.8    | 0.9451800  | 10829.24    | 1377.44 | 14.57% | 0.942156724 | 9421.57   | (30.23)  | 0.32% |
| 10/28/2009         | 9532.89   | 0.9532890  | 10829.24    | 1296.35 | 13.60% | 0.943370419 | 9433.70   | (99.19)  | 1.04% |
| 10/29/2009         | 9596.93   | 0.9596930  | 10829.24    | 1232.31 | 12.84% | 0.948145704 | 9481.46   | (115.47) | 1.20% |
| 10/30/2009         | 9624.64   | 0.9624640  | 10829.24    | 1204.60 | 12.52% | 0.953482812 | 9534.83   | (89.81)  | 0.93% |
| 10/31/2009         | 9606.15   | 0.9606150  | 10829.24    | 1223.09 | 12.73% | 0.957201168 | 9572.01   | (34.14)  | 0.36% |
|                    |           |            |             |         | 12.75% |             |           |          | 0.45% |

Pada tabel 4.12 dapat dilihat bahwa untuk data nilai tukar rupiah yang *range*-nya cukup besar, skala data sangat berpengaruh dalam melakukan proses prediksi. Ini ditunjukkan dengan nilai MAPE yang berbeda jauh, yakni 12.75% untuk data asli dengan *range* yang besar dan 0.45% untuk data skala.

#### **4.4.3.2 Pengujian dengan Data Asli dan Data Prediksi**

Pengujian ini akan menguji sejauh mana SVM bisa digunakan untuk melakukan prediksi terhadap data beberapa hari ke depan. Pada pengujian ini digunakan data asli dan data hasil prediksi untuk melakukan prediksi untuk beberapa hari ke depan. Pengujian ini menggunakan epsilon SVR dan kernel RBF.

Data harga minyak menggunakan data asli yang memberikan nilai MSE dan MAPE yang kecil sedangkan untuk data nilai tukar Rupiah menggunakan data yang telah dinormalisasi. Data ini menggunakan data 10 *time series*. Untuk tanggal 1 Oktober digunakan 10 data asli, tanggal 2 Oktober menggunakan 9 data asli dan 1 data hasil prediksi, tanggal 3 Oktober menggunakan 8 data asli dan 2 data prediksi dan begitu seterusnya.

Tabel 4.13 Perbandingan prediksi data asli dan data prediksi untuk data harga minyak

| Harga Minyak Asli |           | Diprediksi dengan data asli |          |       | Diprediksi dengan data prediksi |           |        |             |
|-------------------|-----------|-----------------------------|----------|-------|---------------------------------|-----------|--------|-------------|
|                   |           | Epsilon RBF                 |          |       | Epsilon RBF                     |           |        |             |
| Tanggal           | Data Asli | Data Prediksi               | Error    | MAPE  | Data Prediksi                   | Error     | MAPE   |             |
| 10/1/2009         | \$70.82   | 70.15377952                 | (\$0.67) | 0.94% | 70.15377952                     | (\$0.67)  | 0.94%  | data asli   |
| 10/2/2009         | \$69.95   | 71.26851744                 | \$1.32   | 1.88% | 70.68496566                     | \$0.73    | 1.05%  | 9 data asli |
| 10/5/2009         | \$70.41   | 69.95935592                 | (\$0.45) | 0.64% | 70.53111203                     | \$0.12    | 0.17%  | 8 data asli |
| 10/6/2009         | \$70.88   | 70.64902924                 | (\$0.23) | 0.33% | 70.84009969                     | (\$0.04)  | 0.06%  | 7 data asli |
| 10/7/2009         | \$69.57   | 71.06306414                 | \$1.49   | 2.15% | 71.00348237                     | \$1.43    | 2.06%  | 6 data asli |
| 10/8/2009         | \$71.69   | 69.78356662                 | (\$1.91) | 2.66% | 71.08074618                     | (\$0.61)  | 0.85%  | 5 data asli |
| 10/9/2009         | \$71.77   | 71.17767583                 | (\$0.59) | 0.83% | 70.74284035                     | (\$1.03)  | 1.43%  | 4 data asli |
| 10/12/2009        | \$73.27   | 71.89008761                 | (\$1.38) | 1.88% | 70.96973959                     | (\$2.30)  | 3.14%  | 3 data asli |
| 10/13/2009        | \$74.15   | 72.67795054                 | (\$1.47) | 1.99% | 70.52507817                     | (\$3.62)  | 4.89%  | 2 data asli |
| 10/14/2009        | \$75.18   | 73.98923647                 | (\$1.19) | 1.58% | 70.67582151                     | (\$4.50)  | 5.99%  | 1 data asli |
| 10/15/2009        | \$77.58   | 74.95345529                 | (\$2.63) | 3.39% | 70.59624802                     | (\$6.98)  | 9.00%  | 0 data asli |
| 10/16/2009        | \$78.53   | 77.23802212                 | (\$1.29) | 1.65% | 70.6647907                      | (\$7.87)  | 10.02% | 0 data asli |
| 10/19/2009        | \$79.61   | 77.96608349                 | (\$1.64) | 2.06% | 70.62281099                     | (\$8.99)  | 11.29% | 0 data asli |
| 10/20/2009        | \$79.09   | 79.39836376                 | \$0.31   | 0.39% | 70.66962289                     | (\$8.42)  | 10.65% | 0 data asli |
| 10/21/2009        | \$81.37   | 78.59698871                 | (\$2.77) | 3.41% | 70.64282295                     | (\$10.73) | 13.18% | 0 data asli |
| 10/22/2009        | \$81.19   | 80.72357425                 | (\$0.47) | 0.57% | 70.73703814                     | (\$10.45) | 12.87% | 0 data asli |
| 10/23/2009        | \$80.50   | 80.68627645                 | \$0.19   | 0.23% | 70.70018423                     | (\$9.80)  | 12.17% | 0 data asli |
| 10/26/2009        | \$78.68   | 79.93035027                 | \$1.25   | 1.59% | 70.78009063                     | (\$7.90)  | 10.04% | 0 data asli |
| 10/27/2009        | \$79.55   | 78.51282414                 | (\$1.04) | 1.30% | 70.76549789                     | (\$8.78)  | 11.04% | 0 data asli |
| 10/28/2009        | \$77.46   | 78.7609601                  | \$1.30   | 1.68% | 70.80542545                     | (\$6.65)  | 8.59%  | 0 data asli |
| 10/29/2009        | \$79.87   | 77.35751117                 | (\$2.51) | 3.15% | 70.79936788                     | (\$9.07)  | 11.36% | 0 data asli |
|                   |           |                             |          | 1.63% |                                 |           | 6.70%  |             |

Dari hasil pengujian pada tabel 4.13 terlihat bahwa untuk lima hari ke depan, untuk data harga minyak SVM bisa memberikan prediksi dengan gabungan data asli dan data prediksi yang cukup baik yang ditunjukkan dengan nilai MAPE yang hampir sama seperti MAPE pada prediksi dengan data asli.

Tabel 4.14 Perbandingan prediksi data asli dan data prediksi untuk data nilai tukar Rupiah terhadap US Dollar

| Nilai Tukar Rupiah Skala |           |            | Diprediksi dengan data asli |           |          |       | Diprediksi dengan data prediksi |           |         |       |             |
|--------------------------|-----------|------------|-----------------------------|-----------|----------|-------|---------------------------------|-----------|---------|-------|-------------|
| Nilai Tukar Rupiah Skala |           |            | E RBF                       |           |          |       | E RBF                           |           |         |       |             |
| Tanggal                  | Data Asli | Data Skala | Data Skala                  | Data Asli | Error    | MAPE  | Data Skala                      | Data Asli | Error   | MAPE  |             |
| 10/1/2009                | 9661.84   | 0.966184   | 0.9720678                   | 9720.68   | 58.84    | 0.61% | 0.9720678                       | 9720.68   | 58.838  | 0.61% | data asli   |
| 10/2/2009                | 9699.32   | 0.969932   | 0.9689453                   | 9689.45   | (9.87)   | 0.10% | 0.9717753                       | 9717.75   | 18.433  | 0.19% | 9 data asli |
| 10/3/2009                | 9699.32   | 0.969932   | 0.9694994                   | 9694.99   | (4.33)   | 0.04% | 0.9716391                       | 9716.39   | 17.071  | 0.18% | 8 data asli |
| 10/4/2009                | 9643.2    | 0.964320   | 0.9698799                   | 9698.80   | 55.60    | 0.58% | 0.9716596                       | 9716.60   | 73.396  | 0.76% | 7 data asli |
| 10/5/2009                | 9643.2    | 0.964320   | 0.9672871                   | 9672.87   | 29.67    | 0.31% | 0.9718441                       | 9718.44   | 75.241  | 0.78% | 6 data asli |
| 10/6/2009                | 9615.38   | 0.961538   | 0.9658677                   | 9658.68   | 43.30    | 0.45% | 0.9717700                       | 9717.70   | 102.320 | 1.06% | 5 data asli |
| 10/7/2009                | 9551.1    | 0.955110   | 0.9638406                   | 9638.41   | 87.31    | 0.91% | 0.9716607                       | 9716.61   | 165.507 | 1.73% | 4 data asli |
| 10/8/2009                | 9532.89   | 0.953289   | 0.9598645                   | 9598.65   | 65.76    | 0.69% | 0.9716069                       | 9716.07   | 183.179 | 1.92% | 3 data asli |
| 10/9/2009                | 9398.5    | 0.939850   | 0.9567998                   | 9568.00   | 169.50   | 1.80% | 0.9714188                       | 9714.19   | 315.688 | 3.36% | 2 data asli |
| 10/10/2009               | 9425.07   | 0.942507   | 0.9487175                   | 9487.18   | 62.11    | 0.66% | 0.9712317                       | 9712.32   | 287.247 | 3.05% | 1 data asli |
| 10/11/2009               | 9425.07   | 0.942507   | 0.9464830                   | 9464.83   | 39.76    | 0.42% | 0.9711734                       | 9711.73   | 286.664 | 3.04% | 0 data asli |
| 10/12/2009               | 9425.07   | 0.942507   | 0.9448298                   | 9448.30   | 23.23    | 0.25% | 0.9710819                       | 9710.82   | 285.749 | 3.03% | 0 data asli |
| 10/13/2009               | 9398.5    | 0.939850   | 0.9434365                   | 9434.37   | 35.87    | 0.38% | 0.9709881                       | 9709.88   | 311.381 | 3.31% | 0 data asli |
| 10/14/2009               | 9363.3    | 0.936330   | 0.9414479                   | 9414.48   | 51.18    | 0.55% | 0.9708852                       | 9708.85   | 345.552 | 3.69% | 0 data asli |
| 10/15/2009               | 9345.79   | 0.934579   | 0.9385303                   | 9385.30   | 39.51    | 0.42% | 0.9707709                       | 9707.71   | 361.919 | 3.87% | 0 data asli |
| 10/16/2009               | 9363.3    | 0.936330   | 0.9367337                   | 9367.34   | 4.04     | 0.04% | 0.9706625                       | 9706.63   | 343.325 | 3.67% | 0 data asli |
| 10/17/2009               | 9372.07   | 0.937207   | 0.9367209                   | 9367.21   | (4.86)   | 0.05% | 0.9705584                       | 9705.58   | 333.514 | 3.56% | 0 data asli |
| 10/18/2009               | 9398.5    | 0.939850   | 0.9368764                   | 9368.76   | (29.74)  | 0.32% | 0.9704522                       | 9704.52   | 306.022 | 3.26% | 0 data asli |
| 10/19/2009               | 9398.5    | 0.939850   | 0.9384233                   | 9384.23   | (14.27)  | 0.15% | 0.9703504                       | 9703.50   | 305.004 | 3.25% | 0 data asli |
| 10/20/2009               | 9389.67   | 0.938967   | 0.9387912                   | 9387.91   | (1.76)   | 0.02% | 0.9702501                       | 9702.50   | 312.831 | 3.33% | 0 data asli |
| 10/21/2009               | 9389.67   | 0.938967   | 0.9386559                   | 9386.56   | (3.11)   | 0.03% | 0.9701470                       | 9701.47   | 311.800 | 3.32% | 0 data asli |
| 10/22/2009               | 9354.54   | 0.935454   | 0.9387712                   | 9387.71   | 33.17    | 0.35% | 0.9700446                       | 9700.45   | 345.906 | 3.70% | 0 data asli |
| 10/23/2009               | 9372.07   | 0.937207   | 0.9372871                   | 9372.87   | 0.80     | 0.01% | 0.9699425                       | 9699.43   | 327.355 | 3.49% | 0 data asli |
| 10/24/2009               | 9425.07   | 0.942507   | 0.9375841                   | 9375.84   | (49.23)  | 0.52% | 0.9698411                       | 9698.41   | 273.341 | 2.90% | 0 data asli |
| 10/25/2009               | 9433.96   | 0.943396   | 0.9402099                   | 9402.10   | (31.86)  | 0.34% | 0.9697406                       | 9697.41   | 263.446 | 2.79% | 0 data asli |
| 10/26/2009               | 9433.96   | 0.943396   | 0.9415737                   | 9415.74   | (18.22)  | 0.19% | 0.9696405                       | 9696.41   | 262.445 | 2.78% | 0 data asli |
| 10/27/2009               | 9451.8    | 0.945180   | 0.9421567                   | 9421.57   | (30.23)  | 0.32% | 0.9695408                       | 9695.41   | 243.608 | 2.58% | 0 data asli |
| 10/28/2009               | 9532.89   | 0.953289   | 0.9433704                   | 9433.70   | (99.19)  | 1.04% | 0.9694416                       | 9694.42   | 161.526 | 1.69% | 0 data asli |
| 10/29/2009               | 9596.93   | 0.959693   | 0.9481457                   | 9481.46   | (115.47) | 1.20% | 0.969342                        | 9693.43   | 96.498  | 1.01% | 0 data asli |
| 10/30/2009               | 9624.64   | 0.962464   | 0.9534828                   | 9534.83   | (89.81)  | 0.93% | 0.9692443                       | 9692.44   | 67.803  | 0.70% | 0 data asli |
| 10/31/2009               | 9606.15   | 0.960615   | 0.9572011                   | 9572.01   | (34.14)  | 0.36% | 0.9691464                       | 9691.46   | 85.314  | 0.89% | 0 data asli |
|                          |           |            |                             |           |          | 0.45% |                                 |           |         | 2.37% |             |

Dari hasil pengujian pada tabel 4.14, sama seperti pada data harga minyak, untuk prediksi lima hari ke depan SVM bisa memberikan hasil prediksi nilai tukar Rupiah yang cukup baik yang ditunjukkan dengan nilai MAPE yang hampir sama seperti nilai MAPE pada prediksi data asli.

Secara umum, pada 10 data *time series*, SVM bisa memberikan hasil prediksi yang cukup baik untuk lima hari ke depan.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil penelitian, analisis dan pembangunan sistem *Support Vector Regression* dengan menggunakan metode *Support Vector Machine* yaitu SVReg dan pengujian SVReg, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. *Support Vector Machine* dapat menjadi alternatif metode untuk melakukan prediksi data *time series*.
2. Prediksi dengan *Support Vector Machine* bisa menghasilkan prediksi untuk lima hari ke depan dengan 10 data *time series* dengan nilai MAPE berkisar di bawah 3%.
3. Data yang digunakan dalam proses pelatihan maupun prediksi mempunyai pengaruh yang cukup besar terhadap proses prediksi. Untuk itu diperlukan normalisasi data untuk data dengan *noise* yang cukup tinggi.
4. Format data set pelatihan dan data set pengujian harus sama, termasuk jumlah data *time series* yang digunakan dan cara normalisasi.
5. Pemilihan parameter C, epsilon dan kernel dalam pembentukan model juga mempunyai pengaruh yang cukup besar dalam proses prediksi.

6. Penggunaan MSE tidak bisa digunakan untuk perbandingan nilai error pada data set yang berbeda karena nilainya yang dependen terhadap data yang diprediksi. MAPE menjadi solusi untuk mengatasi masalah ini karena penggunaan nilai persentase pada MAPE membuat ukuran error menjadi independen.

## 5.2. Saran

Saran yang dapat diberikan untuk pengembangan sistem *Support Vector Regression* yang menerapkan metode *Support Vector Machine* selanjutnya :

1. Penggunaan data lain untuk menguji tingkat akurasi *Support Vector Machine* dalam melakukan prediksi.
2. Normalisasi data yang lebih baik untuk mendapatkan data dengan *noise* yang kecil atau tidak ada sama sekali.
3. Pemilihan parameter yang lebih baik untuk membangun model SVM yang lebih baik.

## DAFTAR PUSTAKA

- Afiahayati, 2008, *Multiple Sequence Alignment Menggunakan Hidden Markov Model*, FMIPA UGM, Yogyakarta.
- Alexandridis, A. and Livanis, E., 2008, *Forecasting Crude Oil Prices Using Wavelet Neural Networks*, Departemen of Accounting and Finance, University of Macedonia of Economics and Social Studies, Thessaloniki.
- Ali, S. and Smith, K.A., 2003, Automatic Parameter Selection for Polynomial Kernel, *International Conference on Information Reuse and Integration*, IRI 2003, 27-29 Oct. 2003, IEEE: 243 – 249.
- Anggraini, S., 2009, *Peramalan Permintaan Uang Cash di ATM Menggunakan Fusi Multi Model dengan Ensemble GRNN*, FMIPA UGM, Yogyakarta.
- Anonim, 2003, Support Vector Machine Regression, <http://kernelsvm.tripod.com>, diakses tanggal 4 Maret 2009.
- Athanasios, S. and Costas, S., 2005, *Time Series Forecasting of Averaged Data With Efficient Use of Information*, IEEE Transactions on Systems, man, and Cybernetics - Part A: Systems and Humans, Vol. 35, No. 5.
- Booch, G., Rumbaugh, J. and Jacobson, I., 2005, *The Unified Modeling Language User Guide*. Addison Wesley Professional.
- Burges, C. J. C., 1998, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, 2(2):121-167.
- Cao, L. J. and Francis, E. H. T., 2003, *Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting*, The National University of Singapore, Singapore 119260, IEEE Transactions on Neural Networks, Vol. 14, No. 6.
- Chang, C.C. and Lin, C.J., 2009, LIBSVM -- A Library for Support Vector Machines, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, diakses tanggal 22 Juni 2009



- Chen, B.J., Chang, M.W. and Lin, C.J., 2004, *Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001*, IEEE Transactions on Power Systems, Volume 19(4): 1821-1830
- Cogger, K.O. and Fanning, K., 1997, *An Introduction to Adaptive Logic Networks With an Application to Audit Risk Assessment*, American Accounting Association, Dallas.
- Christianini, N. and Shawetaylor, J., 2000, *An Introduction to Support Vector Machines*, Cambridge University Press, London.
- Darmadi, K., Chein, I. C. and Hanseok, K., 1994, *Neural Network Technology For Stock Market Index Prediction*, Department of Electrical Engineering, University of Maryland Baltimore County Baltimore.
- DTREG Software For Predictive Modeling and Forecasting , <http://dtreg.com>, diakses tanggal 1 September 2009.
- Frontzek, T., Navin Lal, T. and Eckmiller, R., 2001, *Predicting the Nonlinear Dynamics of Biological Neurons using Support Vector Machines with Different Kernels*, Departement of Computer Science, University of Bonn, Bonn.
- Gao, R. and Tsoukalas, L.H., 2001, *Neural-wavelet Methodology for Load Forecasting*, Journal of Intelligent & Robotic Systems vol. 31, pp. 149-157.
- Hsu, C.W., Chang, C.C. and Lin, C.J., 2003, *A Practical Guide to Support Vector Clasification*, Departement of Computer Science, National Taiwan University, Taipei.
- Hsuan, T.L., 2009, Does Libsvm Require Data Scaling, <http://agbs.kyb.tuebingen.mpg.de/km/bb/showthread.php?tid=1212> , diakses tanggal 10 Februari 2009
- Johan, A. K. S., 2001, *Nonlinear Modeling and Support Vector Machines*, Department of Electrical Engineering ESAT-SISTA, Belgium.
- Karatzoglou, A., Meyer, D. and Hornik, K., 2006, *Support Vector Machine in R*, Journal of Statistical Software 15(9).

- Kim, T. W. and Valdes, J. B., 2003, *Nonlinear Model for Drought Forecasting Based On a Conjunction of Wavelet Transforms and Neural Networks*, Journal of Hydrologic Engineering, 8:6 (319-328).
- Lin, H. T. and Lin, C. J., 2003, A study On Sigmoid Kernels for Support Vector Machine and the Training of non-PSD Kernels by SMO-type methods, <http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf>, diakses tanggal 16 Januari 2009.
- Lu, W. Z., Wang, W. J., Leung, A. Y. T., Lo, S. M., Yuen, R. K. K., Xu, Z. and Fan, H.Y., 2002, Air Pollutant Parameter Forecasting Using Support Vector Machines, *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002. IJCNN '02. , May 12-17, 2002. IEEE, Volume: 1, Pages: 630 – 635.
- Lung, C.C., 2006, *Finding Kernel Function for Stock Market Prediction with Support Vector Regression*, Faculty of Computer Science and Information System, University Technology Malaysia.
- Mukherjee, S., Osuna, E. and Girosi, F., 1997, *Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines*, Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge.
- Muller, J. T., Smola, A. J., Ratsh, G., Scholkopf, B., Kohlmorgen, J. and V. Vapnik, 2000, *Using Support Vector Machines for Time Series Prediction*.
- Muller, J. T., Smola, A.J., Ratsh, G., Scholkopf, B., Kohlmorgen, J. and V.Vapnik, 1997, *Predicting Time Series With Support Vector Machines*, In ICANN, pages 999-1004.
- Ng, H.S. and Lam, K. P., 2000, *Modeling of NASDAQ-GEM Stock Price Relationship Using Neural Network*, Department of System Engineering & 152 Engineering Management, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong.
- Nor, I.A.N, Harun, S. and Kassim, A.H.M., 2007, *Radial Basis Function Modelling of Hourly Streamflow Hydrograph*, Journal of Hydrologic Engineering, ASCE: 113-123.

- Nugroho, A.S., 2006, Apakah Support Vector Machine Itu, <http://asnugroho.wordpress.com/2006/10/15/apakah-support-vector-machine-itu/>, diakses tanggal 10 Februari 2009.
- Nugroho, A.S., 2008, Data Scaling dalam SVM, <http://asnugroho.wordpress.com> , diakses tanggal 29 Oktober 2009.
- Nugroho, A.S., Witarto, A.B. dan Handoko, B., 2003, Support Vector Machine – Teori dan Aplikasinya dalam Bioinformatika1–, <http://ilmukomputer.com>.
- OANDA, The Currency Site : Foreign Exchange Services and Trading, <http://www.oanda.com>, diakses tanggal 2 November 2009.
- Palit, A.K. and Popovic, D., 2005, *Computational Intelligence in Time Series Forecasting : Theory and Engineering Applications*, Springer-Verlag, London.
- Prasetyo, P.K., 2006, Training set, Testing set, dan Validation set, <http://philips.wordpress.com/2006/08/10/training-set-testing-set-dan-validation-set/>, diakses tanggal 10 Maret 2009.
- Rojas, I. and Pomares, H., 2004, Soft Computing Techniques for Time Series Forecasting, *Proceedings European Symposium on Artificial Neural Networks* Bruges (Belgium), 28-30 April 2004, 93-102.
- Sarle, W.S., 2002, Neural Network FAQ, [ftp://ftp.sas.com/pub/neural/FAQ2.html#A\\_sta](ftp://ftp.sas.com/pub/neural/FAQ2.html#A_sta), diakses tanggal 10 Februari 2009.
- Schmidt,J.W., Matthes,F. and Niederée,C., 1999, *Object-Oriented Analysis and Design Course Lecture Slide*, TU Hamburg, Hamburg.
- Scholkopf, B. and Smola, A. J., 2002, *Learning with Kernels*, MIT Press, Massachusetts.
- Sembiring, K., 2007, *Tutorial SVM Bahasa Indonesia*, Sekolah Teknik Elektro dan Informatika, ITB, Bandung.
- Smola, A. J. and Scholkopf, B., 1998, *A Tutorial on Support Vector Regression*, Technical Report NC2-TR-1998-030, NeuroCOLT2.
- Tay, E. H. and Cao, L. J., 2001, *Application of Support Vector Machines to Financial Time Series Forecasting*. Omega, 29:309-317.

- Theodore, B. T. and Huseyin, I., 2000, *Support Vector Machine for Regression And Applications To Financial Forecasting*, University of Oklahoma, Norman, Oklahoma.
- Union Pacific, <http://www.up.com>, diakses tanggal 2 November 2009.
- Valverde, R. M. C., Ferreira, N. J. and Velho, H. F., 2005, *Artificial Neural Network Technique for Rainfall Forecasting Applied To the Sao Paulo Region*, Journal of Hydrology. 301:146–162.
- Vanajakshi, L. and Rilett, L.R., 2004, *A Comparison of The Performance of Artificial Neural Network and Support Vector Machines for The Prediction of Traffic Speed*, 2004 IEEE Intelligent Vehicles Symposium, June 14-17, University of Parma, Italy:IEEE, 194-199.
- Vapnik, V., 1995, *The Nature of Statisticak Learning Theory*, Springer-Verlag New York, Inc.
- Vapnik, V., 1998, *Statistical Learning Theory*, New York: Wiley.
- Vapnik, V., Golowich, S.E. and Smola, A.J., 1996, *Support Vector Method for Function Approximation, Regression Estimation and Signal Processing*, NIPS 1996: 281-287
- Wang, S., Lean, Y.U. and Lai, K.K., 2004, *Crude oil Price Forecasting with TEI@I Methodology*, Journal of System Science and Complexity, Vol. 18 No. 02.
- Wibowo, B.Y., 2003, *Implementasi Jaringan Syaraf Tiruan Metode Feedforward/Backpropagation untuk Memprediksi Harga Saham*, FMIPA UGM, Yogyakarta.
- Wong, S.F. and Wong, K.Y.K., 2004, *Wavelet for Nonlinear Regression using Probabilistic Framework*, Departemen of Computer Sicence and Information Systems, The University of Hong Kong, Hong Kong.
- Wu, C. H., Ho, J. M. and Lee, D. T., 2004, *Travel Time Prediction With Support Vector Regression*, IEEE Transaction on Intelligent Transportation Systems, Volume 5(4): 276-281.

- Wu, X. and Kumar, V., 2007, *Top 10 Algorithms in Data Mining*, Springer-Verlag, London.
- Xie, W., Yu, L. and Xu, S., 2006, *A New Method for Crude Oil Price Forecasting Based on Support Vector Machines*, Institute of System Science, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing.
- Zhu, J.Y., Ren, B., Zhang, H. X. and Deng, Z. T., 2002, Time series prediction via new support vector machines, *Proceedings of The First International Conference on Machine Learning and Cybernetics*, November 4-5 2002, Beijing:IEEE, 364-366.

**LAMPIRAN I**  
**DATA SET HARGA MINYAK**

**Data set pelatihan harga minyak (1 Januari 2009-30 September 2009)**

|       |         |         |         |         |         |         |         |         |         |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 36.51 | 1:46.34 | 2:48.81 | 3:48.58 | 4:42.63 | 5:41.70 | 6:40.83 | 7:37.59 | 8:37.78 | 9:37.28 | 10:35.40 |
| 38.74 | 1:48.81 | 2:48.58 | 3:42.63 | 4:41.70 | 5:40.83 | 6:37.59 | 7:37.78 | 8:37.28 | 9:35.40 | 10:36.51 |
| 42.25 | 1:48.58 | 2:42.63 | 3:41.70 | 4:40.83 | 5:37.59 | 6:37.78 | 7:37.28 | 8:35.40 | 9:36.51 | 10:38.74 |
| 42.27 | 1:42.63 | 2:41.70 | 3:40.83 | 4:37.59 | 5:37.78 | 6:37.28 | 7:35.40 | 8:36.51 | 9:38.74 | 10:42.25 |
| 45.47 | 1:41.70 | 2:40.83 | 3:37.59 | 4:37.78 | 5:37.28 | 6:35.40 | 7:36.51 | 8:38.74 | 9:42.25 | 10:42.27 |
| 45.73 | 1:40.83 | 2:37.59 | 3:37.78 | 4:37.28 | 5:35.40 | 6:36.51 | 7:38.74 | 8:42.25 | 9:42.27 | 10:45.47 |
| 41.58 | 1:37.59 | 2:37.78 | 3:37.28 | 4:35.40 | 5:36.51 | 6:38.74 | 7:42.25 | 8:42.27 | 9:45.47 | 10:45.73 |
| 42.16 | 1:37.78 | 2:37.28 | 3:35.40 | 4:36.51 | 5:38.74 | 6:42.25 | 7:42.27 | 8:45.47 | 9:45.73 | 10:41.58 |
| 41.44 | 1:37.28 | 2:35.40 | 3:36.51 | 4:38.74 | 5:42.25 | 6:42.27 | 7:45.47 | 8:45.73 | 9:41.58 | 10:42.16 |
| 41.68 | 1:35.40 | 2:36.51 | 3:38.74 | 4:42.25 | 5:42.27 | 6:45.47 | 7:45.73 | 8:41.58 | 9:42.16 | 10:41.44 |
| 40.08 | 1:36.51 | 2:38.74 | 3:42.25 | 4:42.27 | 5:45.47 | 6:45.73 | 7:41.58 | 8:42.16 | 9:41.44 | 10:41.68 |
| 40.78 | 1:38.74 | 2:42.25 | 3:42.27 | 4:45.47 | 5:45.73 | 6:41.58 | 7:42.16 | 8:41.44 | 9:41.68 | 10:40.08 |
| 40.32 | 1:42.25 | 2:42.27 | 3:45.47 | 4:45.73 | 5:41.58 | 6:42.16 | 7:41.44 | 8:41.68 | 9:40.08 | 10:40.78 |
| 41.17 | 1:42.27 | 2:45.47 | 3:45.73 | 4:41.58 | 5:42.16 | 6:41.44 | 7:41.68 | 8:40.08 | 9:40.78 | 10:40.32 |
| 40.17 | 1:45.47 | 2:45.73 | 3:41.58 | 4:42.16 | 5:41.44 | 6:41.68 | 7:40.08 | 8:40.78 | 9:40.32 | 10:41.17 |
| 39.56 | 1:45.73 | 2:41.58 | 3:42.16 | 4:41.44 | 5:41.68 | 6:40.08 | 7:40.78 | 8:40.32 | 9:41.17 | 10:40.17 |
| 37.55 | 1:41.58 | 2:42.16 | 3:41.44 | 4:41.68 | 5:40.08 | 6:40.78 | 7:40.32 | 8:41.17 | 9:40.17 | 10:39.56 |
| 35.94 | 1:42.16 | 2:41.44 | 3:41.68 | 4:40.08 | 5:40.78 | 6:40.32 | 7:41.17 | 8:40.17 | 9:39.56 | 10:37.55 |
| 33.98 | 1:41.44 | 2:41.68 | 3:40.08 | 4:40.78 | 5:40.32 | 6:41.17 | 7:40.17 | 8:39.56 | 9:37.55 | 10:35.94 |
| 37.51 | 1:41.68 | 2:40.08 | 3:40.78 | 4:40.32 | 5:41.17 | 6:40.17 | 7:39.56 | 8:37.55 | 9:35.94 | 10:33.98 |
| 34.93 | 1:40.08 | 2:40.78 | 3:40.32 | 4:41.17 | 5:40.17 | 6:39.56 | 7:37.55 | 8:35.94 | 9:33.98 | 10:37.51 |
| 34.62 | 1:40.78 | 2:40.32 | 3:41.17 | 4:40.17 | 5:39.56 | 6:37.55 | 7:35.94 | 8:33.98 | 9:37.51 | 10:34.93 |
| 39.48 | 1:40.32 | 2:41.17 | 3:40.17 | 4:39.56 | 5:37.55 | 6:35.94 | 7:33.98 | 8:37.51 | 9:34.93 | 10:34.62 |
| 38.94 | 1:41.17 | 2:40.17 | 3:39.56 | 4:37.55 | 5:35.94 | 6:33.98 | 7:37.51 | 8:34.93 | 9:34.62 | 10:39.48 |
| 37.59 | 1:40.17 | 2:39.56 | 3:37.55 | 4:35.94 | 5:33.98 | 6:37.51 | 7:34.93 | 8:34.62 | 9:39.48 | 10:38.94 |
| 38.96 | 1:39.56 | 2:37.55 | 3:35.94 | 4:33.98 | 5:37.51 | 6:34.93 | 7:34.62 | 8:39.48 | 9:38.94 | 10:37.59 |
| 42.50 | 1:37.55 | 2:35.94 | 3:33.98 | 4:37.51 | 5:34.93 | 6:34.62 | 7:39.48 | 8:38.94 | 9:37.59 | 10:38.96 |
| 45.22 | 1:35.94 | 2:33.98 | 3:37.51 | 4:34.93 | 5:34.62 | 6:39.48 | 7:38.94 | 8:37.59 | 9:38.96 | 10:42.50 |
| 44.76 | 1:33.98 | 2:37.51 | 3:34.93 | 4:34.62 | 5:39.48 | 6:38.94 | 7:37.59 | 8:38.96 | 9:42.50 | 10:45.22 |
| 40.15 | 1:37.51 | 2:34.93 | 3:34.62 | 4:39.48 | 5:38.94 | 6:37.59 | 7:38.96 | 8:42.50 | 9:45.22 | 10:44.76 |
| 41.65 | 1:34.93 | 2:34.62 | 3:39.48 | 4:38.94 | 5:37.59 | 6:38.96 | 7:42.50 | 8:45.22 | 9:44.76 | 10:40.15 |
| 45.38 | 1:34.62 | 2:39.48 | 3:38.94 | 4:37.59 | 5:38.96 | 6:42.50 | 7:45.22 | 8:44.76 | 9:40.15 | 10:41.65 |
| 43.61 | 1:39.48 | 2:38.94 | 3:37.59 | 4:38.96 | 5:42.50 | 6:45.22 | 7:44.76 | 8:40.15 | 9:41.65 | 10:45.38 |
| 45.52 | 1:38.94 | 2:37.59 | 3:38.96 | 4:42.50 | 5:45.22 | 6:44.76 | 7:40.15 | 8:41.65 | 9:45.38 | 10:43.61 |
| 47.07 | 1:37.59 | 2:38.96 | 3:42.50 | 4:45.22 | 5:44.76 | 6:40.15 | 7:41.65 | 8:45.38 | 9:43.61 | 10:45.52 |
| 45.71 | 1:38.96 | 2:42.50 | 3:45.22 | 4:44.76 | 5:40.15 | 6:41.65 | 7:45.38 | 8:43.61 | 9:45.52 | 10:47.07 |
| 42.33 | 1:42.50 | 2:45.22 | 3:44.76 | 4:40.15 | 5:41.65 | 6:45.38 | 7:43.61 | 8:45.52 | 9:47.07 | 10:45.71 |
| 47.03 | 1:45.22 | 2:44.76 | 3:40.15 | 4:41.65 | 5:45.38 | 6:43.61 | 7:45.52 | 8:47.07 | 9:45.71 | 10:42.33 |
| 46.25 | 1:44.76 | 2:40.15 | 3:41.65 | 4:45.38 | 5:43.61 | 6:45.52 | 7:47.07 | 8:45.71 | 9:42.33 | 10:47.03 |
| 47.35 | 1:40.15 | 2:41.65 | 3:45.38 | 4:43.61 | 5:45.52 | 6:47.07 | 7:45.71 | 8:42.33 | 9:47.03 | 10:46.25 |
| 49.16 | 1:41.65 | 2:45.38 | 3:43.61 | 4:45.52 | 5:47.07 | 6:45.71 | 7:42.33 | 8:47.03 | 9:46.25 | 10:47.35 |
| 48.14 | 1:45.38 | 2:43.61 | 3:45.52 | 4:47.07 | 5:45.71 | 6:42.33 | 7:47.03 | 8:46.25 | 9:47.35 | 10:49.16 |
| 51.61 | 1:43.61 | 2:45.52 | 3:47.07 | 4:45.71 | 5:42.33 | 6:47.03 | 7:46.25 | 8:47.35 | 9:49.16 | 10:48.14 |
| 51.06 | 1:45.52 | 2:47.07 | 3:45.71 | 4:42.33 | 5:47.03 | 6:46.25 | 7:47.35 | 8:49.16 | 9:48.14 | 10:51.61 |
| 53.00 | 1:47.07 | 2:45.71 | 3:42.33 | 4:47.03 | 5:46.25 | 6:47.35 | 7:49.16 | 8:48.14 | 9:51.61 | 10:51.06 |
| 53.48 | 1:45.71 | 2:42.33 | 3:47.03 | 4:46.25 | 5:47.35 | 6:49.16 | 7:48.14 | 8:51.61 | 9:51.06 | 10:53.00 |
| 52.27 | 1:42.33 | 2:47.03 | 3:46.25 | 4:47.35 | 5:49.16 | 6:48.14 | 7:51.61 | 8:51.06 | 9:53.00 | 10:53.48 |
| 54.34 | 1:47.03 | 2:46.25 | 3:47.35 | 4:49.16 | 5:48.14 | 6:51.61 | 7:51.06 | 8:53.00 | 9:53.48 | 10:52.27 |
| 52.38 | 1:46.25 | 2:47.35 | 3:49.16 | 4:48.14 | 5:51.61 | 6:51.06 | 7:53.00 | 8:53.48 | 9:52.27 | 10:54.34 |
| 48.41 | 1:47.35 | 2:49.16 | 3:48.14 | 4:51.61 | 5:51.06 | 6:53.00 | 7:53.48 | 8:52.27 | 9:54.34 | 10:52.38 |
| 49.66 | 1:49.16 | 2:48.14 | 3:51.61 | 4:51.06 | 5:53.00 | 6:53.48 | 7:52.27 | 8:54.34 | 9:52.38 | 10:48.41 |
| 48.39 | 1:48.14 | 2:51.61 | 3:51.06 | 4:53.00 | 5:53.48 | 6:52.27 | 7:54.34 | 8:52.38 | 9:48.41 | 10:49.66 |
| 52.64 | 1:51.61 | 2:51.06 | 3:53.00 | 4:53.48 | 5:52.27 | 6:54.34 | 7:52.38 | 8:48.41 | 9:49.66 | 10:48.39 |
| 52.51 | 1:51.06 | 2:53.00 | 3:53.48 | 4:52.27 | 5:54.34 | 6:52.38 | 7:48.41 | 8:49.66 | 9:48.39 | 10:52.64 |

|       |         |         |         |         |         |         |         |         |         |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 51.05 | 1:53.00 | 2:53.48 | 3:52.27 | 4:54.34 | 5:52.38 | 6:48.41 | 7:49.66 | 8:48.39 | 9:52.64 | 10:52.51 |
| 49.15 | 1:53.48 | 2:52.27 | 3:54.34 | 4:52.38 | 5:48.41 | 6:49.66 | 7:48.39 | 8:52.64 | 9:52.51 | 10:51.05 |
| 49.38 | 1:52.27 | 2:54.34 | 3:52.38 | 4:48.41 | 5:49.66 | 6:48.39 | 7:52.64 | 8:52.51 | 9:51.05 | 10:49.15 |
| 52.24 | 1:54.34 | 2:52.38 | 3:48.41 | 4:49.66 | 5:48.39 | 6:52.64 | 7:52.51 | 8:51.05 | 9:49.15 | 10:49.38 |
| 50.05 | 1:52.38 | 2:48.41 | 3:49.66 | 4:48.39 | 5:52.64 | 6:52.51 | 7:51.05 | 8:49.15 | 9:49.38 | 10:52.24 |
| 49.41 | 1:48.41 | 2:49.66 | 3:48.39 | 4:52.64 | 5:52.51 | 6:51.05 | 7:49.15 | 8:49.38 | 9:52.24 | 10:50.05 |
| 49.25 | 1:49.66 | 2:48.39 | 3:52.64 | 4:52.51 | 5:51.05 | 6:49.15 | 7:49.38 | 8:52.24 | 9:50.05 | 10:49.41 |
| 49.98 | 1:48.39 | 2:52.64 | 3:52.51 | 4:51.05 | 5:49.15 | 6:49.38 | 7:52.24 | 8:50.05 | 9:49.41 | 10:49.25 |
| 50.33 | 1:52.64 | 2:52.51 | 3:51.05 | 4:49.15 | 5:49.38 | 6:52.24 | 7:50.05 | 8:49.41 | 9:49.25 | 10:49.98 |
| 45.88 | 1:52.51 | 2:51.05 | 3:49.15 | 4:49.38 | 5:52.24 | 6:50.05 | 7:49.41 | 8:49.25 | 9:49.98 | 10:50.33 |
| 46.51 | 1:51.05 | 2:49.15 | 3:49.38 | 4:52.24 | 5:50.05 | 6:49.41 | 7:49.25 | 8:49.98 | 9:50.33 | 10:45.88 |
| 47.35 | 1:49.15 | 2:49.38 | 3:52.24 | 4:50.05 | 5:49.41 | 6:49.25 | 7:49.98 | 8:50.33 | 9:45.88 | 10:46.51 |
| 48.62 | 1:49.38 | 2:52.24 | 3:50.05 | 4:49.41 | 5:49.25 | 6:49.98 | 7:50.33 | 8:45.88 | 9:46.51 | 10:47.35 |
| 50.80 | 1:52.24 | 2:50.05 | 3:49.41 | 4:49.25 | 5:49.98 | 6:50.33 | 7:45.88 | 8:46.51 | 9:47.35 | 10:48.62 |
| 50.14 | 1:50.05 | 2:49.41 | 3:49.25 | 4:49.98 | 5:50.33 | 6:45.88 | 7:46.51 | 8:47.35 | 9:48.62 | 10:50.80 |
| 49.92 | 1:49.41 | 2:49.25 | 3:49.98 | 4:50.33 | 5:45.88 | 6:46.51 | 7:47.35 | 8:48.62 | 9:50.80 | 10:50.14 |
| 50.97 | 1:49.25 | 2:49.98 | 3:50.33 | 4:45.88 | 5:46.51 | 6:47.35 | 7:48.62 | 8:50.80 | 9:50.14 | 10:49.92 |
| 51.12 | 1:49.98 | 2:50.33 | 3:45.88 | 4:46.51 | 5:47.35 | 6:48.62 | 7:50.80 | 8:50.14 | 9:49.92 | 10:50.97 |
| 53.20 | 1:50.33 | 2:45.88 | 3:46.51 | 4:47.35 | 5:48.62 | 6:50.80 | 7:50.14 | 8:49.92 | 9:50.97 | 10:51.12 |
| 54.47 | 1:45.88 | 2:46.51 | 3:47.35 | 4:48.62 | 5:50.80 | 6:50.14 | 7:49.92 | 8:50.97 | 9:51.12 | 10:53.20 |
| 53.84 | 1:46.51 | 2:47.35 | 3:48.62 | 4:50.80 | 5:50.14 | 6:49.92 | 7:50.97 | 8:51.12 | 9:53.20 | 10:54.47 |
| 56.34 | 1:47.35 | 2:48.62 | 3:50.80 | 4:50.14 | 5:49.92 | 6:50.97 | 7:51.12 | 8:53.20 | 9:54.47 | 10:53.84 |
| 56.71 | 1:48.62 | 2:50.80 | 3:50.14 | 4:49.92 | 5:50.97 | 6:51.12 | 7:53.20 | 8:54.47 | 9:53.84 | 10:56.34 |
| 58.63 | 1:50.80 | 2:50.14 | 3:49.92 | 4:50.97 | 5:51.12 | 6:53.20 | 7:54.47 | 8:53.84 | 9:56.34 | 10:56.71 |
| 58.50 | 1:50.14 | 2:49.92 | 3:50.97 | 4:51.12 | 5:53.20 | 6:54.47 | 7:53.84 | 8:56.34 | 9:56.71 | 10:58.63 |
| 58.85 | 1:49.92 | 2:50.97 | 3:51.12 | 4:53.20 | 5:54.47 | 6:53.84 | 7:56.34 | 8:56.71 | 9:58.63 | 10:58.50 |
| 58.02 | 1:50.97 | 2:51.12 | 3:53.20 | 4:54.47 | 5:53.84 | 6:56.34 | 7:56.71 | 8:58.63 | 9:58.50 | 10:58.85 |
| 58.62 | 1:51.12 | 2:53.20 | 3:54.47 | 4:53.84 | 5:56.34 | 6:56.71 | 7:58.63 | 8:58.50 | 9:58.85 | 10:58.02 |
| 56.34 | 1:53.20 | 2:54.47 | 3:53.84 | 4:56.34 | 5:56.71 | 6:58.63 | 7:58.50 | 8:58.85 | 9:58.02 | 10:58.62 |
| 59.03 | 1:54.47 | 2:53.84 | 3:56.34 | 4:56.71 | 5:58.63 | 6:58.50 | 7:58.85 | 8:58.02 | 9:58.62 | 10:56.34 |
| 59.65 | 1:53.84 | 2:56.34 | 3:56.71 | 4:58.63 | 5:58.50 | 6:58.85 | 7:58.02 | 8:58.62 | 9:56.34 | 10:59.03 |
| 62.02 | 1:56.34 | 2:56.71 | 3:58.63 | 4:58.50 | 5:58.85 | 6:58.02 | 7:58.62 | 8:56.34 | 9:59.03 | 10:59.65 |
| 60.55 | 1:56.71 | 2:58.63 | 3:58.50 | 4:58.85 | 5:58.02 | 6:58.62 | 7:56.34 | 8:59.03 | 9:59.65 | 10:62.02 |
| 61.17 | 1:58.63 | 2:58.50 | 3:58.85 | 4:58.02 | 5:58.62 | 6:56.34 | 7:59.03 | 8:59.65 | 9:62.02 | 10:60.55 |
| 62.45 | 1:58.50 | 2:58.85 | 3:58.02 | 4:58.62 | 5:56.34 | 6:59.03 | 7:59.65 | 8:62.02 | 9:60.55 | 10:61.17 |
| 63.45 | 1:58.85 | 2:58.02 | 3:58.62 | 4:56.34 | 5:59.03 | 6:59.65 | 7:62.02 | 8:60.55 | 9:61.17 | 10:62.45 |
| 65.08 | 1:58.02 | 2:58.62 | 3:56.34 | 4:59.03 | 5:59.65 | 6:62.02 | 7:60.55 | 8:61.17 | 9:62.45 | 10:63.45 |
| 66.31 | 1:58.62 | 2:56.34 | 3:59.03 | 4:59.65 | 5:62.02 | 6:60.55 | 7:61.17 | 8:62.45 | 9:63.45 | 10:65.08 |
| 68.58 | 1:56.34 | 2:59.03 | 3:59.65 | 4:62.02 | 5:60.55 | 6:61.17 | 7:62.45 | 8:63.45 | 9:65.08 | 10:66.31 |
| 68.55 | 1:59.03 | 2:59.65 | 3:62.02 | 4:60.55 | 5:61.17 | 6:62.45 | 7:63.45 | 8:65.08 | 9:66.31 | 10:68.58 |
| 66.12 | 1:59.65 | 2:62.02 | 3:60.55 | 4:61.17 | 5:62.45 | 6:63.45 | 7:65.08 | 8:66.31 | 9:68.58 | 10:68.55 |
| 68.81 | 1:62.02 | 2:60.55 | 3:61.17 | 4:62.45 | 5:63.45 | 6:65.08 | 7:66.31 | 8:68.58 | 9:68.55 | 10:66.12 |
| 68.44 | 1:60.55 | 2:61.17 | 3:62.45 | 4:63.45 | 5:65.08 | 6:66.31 | 7:68.58 | 8:68.55 | 9:66.12 | 10:68.81 |
| 68.09 | 1:61.17 | 2:62.45 | 3:63.45 | 4:65.08 | 5:66.31 | 6:68.58 | 7:68.55 | 8:66.12 | 9:68.81 | 10:68.44 |
| 70.01 | 1:62.45 | 2:63.45 | 3:65.08 | 4:66.31 | 5:68.58 | 6:68.55 | 7:66.12 | 8:68.81 | 9:68.44 | 10:68.09 |
| 71.33 | 1:63.45 | 2:65.08 | 3:66.31 | 4:68.58 | 5:68.55 | 6:66.12 | 7:68.81 | 8:68.44 | 9:68.09 | 10:70.01 |
| 72.68 | 1:65.08 | 2:66.31 | 3:68.58 | 4:68.55 | 5:66.12 | 6:68.81 | 7:68.44 | 8:68.09 | 9:70.01 | 10:71.33 |
| 72.04 | 1:66.31 | 2:68.58 | 3:68.55 | 4:66.12 | 5:68.81 | 6:68.44 | 7:68.09 | 8:70.01 | 9:71.33 | 10:72.68 |
| 70.62 | 1:68.58 | 2:68.55 | 3:66.12 | 4:68.81 | 5:68.44 | 6:68.09 | 7:70.01 | 8:71.33 | 9:72.68 | 10:72.04 |
| 70.47 | 1:68.55 | 2:66.12 | 3:68.81 | 4:68.44 | 5:68.09 | 6:70.01 | 7:71.33 | 8:72.68 | 9:72.04 | 10:70.62 |
| 71.03 | 1:66.12 | 2:68.81 | 3:68.44 | 4:68.09 | 5:70.01 | 6:71.33 | 7:72.68 | 8:72.04 | 9:70.62 | 10:70.47 |
| 71.37 | 1:68.81 | 2:68.44 | 3:68.09 | 4:70.01 | 5:71.33 | 6:72.68 | 7:72.04 | 8:70.62 | 9:70.47 | 10:71.03 |
| 69.55 | 1:68.44 | 2:68.09 | 3:70.01 | 4:71.33 | 5:72.68 | 6:72.04 | 7:70.62 | 8:70.47 | 9:71.03 | 10:71.37 |
| 66.93 | 1:68.09 | 2:70.01 | 3:71.33 | 4:72.68 | 5:72.04 | 6:70.62 | 7:70.47 | 8:71.03 | 9:71.37 | 10:69.55 |
| 68.79 | 1:70.01 | 2:71.33 | 3:72.68 | 4:72.04 | 5:70.62 | 6:70.47 | 7:71.03 | 8:71.37 | 9:69.55 | 10:66.93 |
| 68.67 | 1:71.33 | 2:72.68 | 3:72.04 | 4:70.62 | 5:70.47 | 6:71.03 | 7:71.37 | 8:69.55 | 9:66.93 | 10:68.79 |
| 70.23 | 1:72.68 | 2:72.04 | 3:70.62 | 4:70.47 | 5:71.03 | 6:71.37 | 7:69.55 | 8:66.93 | 9:68.79 | 10:68.67 |
| 69.16 | 1:72.04 | 2:70.62 | 3:70.47 | 4:71.03 | 5:71.37 | 6:69.55 | 7:66.93 | 8:68.79 | 9:68.67 | 10:70.23 |
| 71.49 | 1:70.62 | 2:70.47 | 3:71.03 | 4:71.37 | 5:69.55 | 6:66.93 | 7:68.79 | 8:68.67 | 9:70.23 | 10:69.16 |
| 69.89 | 1:70.47 | 2:71.03 | 3:71.37 | 4:69.55 | 5:66.93 | 6:68.79 | 7:68.67 | 8:70.23 | 9:69.16 | 10:71.49 |
| 69.31 | 1:71.03 | 2:71.37 | 3:69.55 | 4:66.93 | 5:68.79 | 6:68.67 | 7:70.23 | 8:69.16 | 9:71.49 | 10:69.89 |
| 66.73 | 1:71.37 | 2:69.55 | 3:66.93 | 4:68.79 | 5:68.67 | 6:70.23 | 7:69.16 | 8:71.49 | 9:69.89 | 10:69.31 |
| 64.05 | 1:69.55 | 2:66.93 | 3:68.79 | 4:68.67 | 5:70.23 | 6:69.16 | 7:71.49 | 8:69.89 | 9:69.31 | 10:66.73 |

|       |         |         |         |         |         |         |         |         |         |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 62.93 | 1:66.93 | 2:68.79 | 3:68.67 | 4:70.23 | 5:69.16 | 6:71.49 | 7:69.89 | 8:69.31 | 9:66.73 | 10:64.05 |
| 60.14 | 1:68.79 | 2:68.67 | 3:70.23 | 4:69.16 | 5:71.49 | 6:69.89 | 7:69.31 | 8:66.73 | 9:64.05 | 10:62.93 |
| 60.41 | 1:68.67 | 2:70.23 | 3:69.16 | 4:71.49 | 5:69.89 | 6:69.31 | 7:66.73 | 8:64.05 | 9:62.93 | 10:60.14 |
| 59.89 | 1:70.23 | 2:69.16 | 3:71.49 | 4:69.89 | 5:69.31 | 6:66.73 | 7:64.05 | 8:62.93 | 9:60.14 | 10:60.41 |
| 59.69 | 1:69.16 | 2:71.49 | 3:69.89 | 4:69.31 | 5:66.73 | 6:64.05 | 7:62.93 | 8:60.14 | 9:60.41 | 10:59.89 |
| 59.52 | 1:71.49 | 2:69.89 | 3:69.31 | 4:66.73 | 5:64.05 | 6:62.93 | 7:60.14 | 8:60.41 | 9:59.89 | 10:59.69 |
| 61.54 | 1:69.89 | 2:69.31 | 3:66.73 | 4:64.05 | 5:62.93 | 6:60.14 | 7:60.41 | 8:59.89 | 9:59.69 | 10:59.52 |
| 62.02 | 1:69.31 | 2:66.73 | 3:64.05 | 4:62.93 | 5:60.14 | 6:60.41 | 7:59.89 | 8:59.69 | 9:59.52 | 10:61.54 |
| 63.56 | 1:66.73 | 2:64.05 | 3:62.93 | 4:60.14 | 5:60.41 | 6:59.89 | 7:59.69 | 8:59.52 | 9:61.54 | 10:62.02 |
| 63.98 | 1:64.05 | 2:62.93 | 3:60.14 | 4:60.41 | 5:59.89 | 6:59.69 | 7:59.52 | 8:61.54 | 9:62.02 | 10:63.56 |
| 64.72 | 1:62.93 | 2:60.14 | 3:60.41 | 4:59.89 | 5:59.69 | 6:59.52 | 7:61.54 | 8:62.02 | 9:63.56 | 10:63.98 |
| 63.57 | 1:60.14 | 2:60.41 | 3:59.89 | 4:59.69 | 5:59.52 | 6:61.54 | 7:62.02 | 8:63.56 | 9:63.98 | 10:64.72 |
| 66.11 | 1:60.41 | 2:59.89 | 3:59.69 | 4:59.52 | 5:61.54 | 6:62.02 | 7:63.56 | 8:63.98 | 9:64.72 | 10:63.57 |
| 66.55 | 1:59.89 | 2:59.69 | 3:59.52 | 4:61.54 | 5:62.02 | 6:63.56 | 7:63.98 | 8:64.72 | 9:63.57 | 10:66.11 |
| 68.38 | 1:59.69 | 2:59.52 | 3:61.54 | 4:62.02 | 5:63.56 | 6:63.98 | 7:64.72 | 8:63.57 | 9:66.11 | 10:66.55 |
| 67.23 | 1:59.52 | 2:61.54 | 3:62.02 | 4:63.56 | 5:63.98 | 6:64.72 | 7:63.57 | 8:66.11 | 9:66.55 | 10:68.38 |
| 63.35 | 1:61.54 | 2:62.02 | 3:63.56 | 4:63.98 | 5:64.72 | 6:63.57 | 7:66.11 | 8:66.55 | 9:68.38 | 10:67.23 |
| 66.94 | 1:62.02 | 2:63.56 | 3:63.98 | 4:64.72 | 5:63.57 | 6:66.11 | 7:66.55 | 8:68.38 | 9:67.23 | 10:63.35 |
| 69.45 | 1:63.56 | 2:63.98 | 3:64.72 | 4:63.57 | 5:66.11 | 6:66.55 | 7:68.38 | 8:67.23 | 9:63.35 | 10:66.94 |
| 71.58 | 1:63.98 | 2:64.72 | 3:63.57 | 4:66.11 | 5:66.55 | 6:68.38 | 7:67.23 | 8:63.35 | 9:66.94 | 10:69.45 |
| 71.42 | 1:64.72 | 2:63.57 | 3:66.11 | 4:66.55 | 5:68.38 | 6:67.23 | 7:63.35 | 8:66.94 | 9:69.45 | 10:71.58 |
| 71.97 | 1:63.57 | 2:66.11 | 3:66.55 | 4:68.38 | 5:67.23 | 6:63.35 | 7:66.94 | 8:69.45 | 9:71.58 | 10:71.42 |
| 71.94 | 1:66.11 | 2:66.55 | 3:68.38 | 4:67.23 | 5:63.35 | 6:66.94 | 7:69.45 | 8:71.58 | 9:71.42 | 10:71.97 |
| 70.93 | 1:66.55 | 2:68.38 | 3:67.23 | 4:63.35 | 5:66.94 | 6:69.45 | 7:71.58 | 8:71.42 | 9:71.97 | 10:71.94 |
| 70.60 | 1:68.38 | 2:67.23 | 3:63.35 | 4:66.94 | 5:69.45 | 6:71.58 | 7:71.42 | 8:71.97 | 9:71.94 | 10:70.93 |
| 69.45 | 1:67.23 | 2:63.35 | 3:66.94 | 4:69.45 | 5:71.58 | 6:71.42 | 7:71.97 | 8:71.94 | 9:70.93 | 10:70.60 |
| 70.16 | 1:63.35 | 2:66.94 | 3:69.45 | 4:71.58 | 5:71.42 | 6:71.97 | 7:71.94 | 8:70.93 | 9:70.60 | 10:69.45 |
| 70.52 | 1:66.94 | 2:69.45 | 3:71.58 | 4:71.42 | 5:71.97 | 6:71.94 | 7:70.93 | 8:70.60 | 9:69.45 | 10:70.16 |
| 67.51 | 1:69.45 | 2:71.58 | 3:71.42 | 4:71.97 | 5:71.94 | 6:70.93 | 7:70.60 | 8:69.45 | 9:70.16 | 10:70.52 |
| 66.75 | 1:71.58 | 2:71.42 | 3:71.97 | 4:71.94 | 5:70.93 | 6:70.60 | 7:69.45 | 8:70.16 | 9:70.52 | 10:67.51 |
| 69.19 | 1:71.42 | 2:71.97 | 3:71.94 | 4:70.93 | 5:70.60 | 6:69.45 | 7:70.16 | 8:70.52 | 9:67.51 | 10:66.75 |
| 72.42 | 1:71.97 | 2:71.94 | 3:70.93 | 4:70.60 | 5:69.45 | 6:70.16 | 7:70.52 | 8:67.51 | 9:66.75 | 10:69.19 |
| 72.54 | 1:71.94 | 2:70.93 | 3:70.60 | 4:69.45 | 5:70.16 | 6:70.52 | 7:67.51 | 8:66.75 | 9:69.19 | 10:72.42 |
| 73.19 | 1:70.93 | 2:70.60 | 3:69.45 | 4:70.16 | 5:70.52 | 6:67.51 | 7:66.75 | 8:69.19 | 9:72.42 | 10:72.54 |
| 73.82 | 1:70.60 | 2:69.45 | 3:70.16 | 4:70.52 | 5:67.51 | 6:66.75 | 7:69.19 | 8:72.42 | 9:72.54 | 10:73.19 |
| 71.55 | 1:69.45 | 2:70.16 | 3:70.52 | 4:67.51 | 5:66.75 | 6:69.19 | 7:72.42 | 8:72.54 | 9:73.19 | 10:73.82 |
| 71.43 | 1:70.16 | 2:70.52 | 3:67.51 | 4:66.75 | 5:69.19 | 6:72.42 | 7:72.54 | 8:73.19 | 9:73.82 | 10:71.55 |
| 72.49 | 1:70.52 | 2:67.51 | 3:66.75 | 4:69.19 | 5:72.42 | 6:72.54 | 7:73.19 | 8:73.82 | 9:71.55 | 10:71.43 |
| 72.79 | 1:67.51 | 2:66.75 | 3:69.19 | 4:72.42 | 5:72.54 | 6:73.19 | 7:73.82 | 8:71.55 | 9:71.43 | 10:72.49 |
| 69.96 | 1:66.75 | 2:69.19 | 3:72.42 | 4:72.54 | 5:73.19 | 6:73.82 | 7:71.55 | 8:71.43 | 9:72.49 | 10:72.79 |
| 68.05 | 1:69.19 | 2:72.42 | 3:72.54 | 4:73.19 | 5:73.82 | 6:71.55 | 7:71.43 | 8:72.49 | 9:72.79 | 10:69.96 |
| 68.05 | 1:72.42 | 2:72.54 | 3:73.19 | 4:73.82 | 5:71.55 | 6:71.43 | 7:72.49 | 8:72.79 | 9:69.96 | 10:68.05 |
| 67.96 | 1:72.54 | 2:73.19 | 3:73.82 | 4:71.55 | 5:71.43 | 6:72.49 | 7:72.79 | 8:69.96 | 9:68.05 | 10:68.05 |
| 68.02 | 1:73.19 | 2:73.82 | 3:71.55 | 4:71.43 | 5:72.49 | 6:72.79 | 7:69.96 | 8:68.05 | 9:68.05 | 10:67.96 |
| 71.10 | 1:73.82 | 2:71.55 | 3:71.43 | 4:72.49 | 5:72.79 | 6:69.96 | 7:68.05 | 8:68.05 | 9:67.96 | 10:68.02 |
| 71.31 | 1:71.55 | 2:71.43 | 3:72.49 | 4:72.79 | 5:69.96 | 6:68.05 | 7:68.05 | 8:67.96 | 9:68.02 | 10:71.10 |
| 71.94 | 1:71.43 | 2:72.49 | 3:72.79 | 4:69.96 | 5:68.05 | 6:68.05 | 7:67.96 | 8:68.02 | 9:71.10 | 10:71.31 |
| 69.29 | 1:72.49 | 2:72.79 | 3:69.96 | 4:68.05 | 5:68.05 | 6:67.96 | 7:68.02 | 8:71.10 | 9:71.31 | 10:71.94 |
| 68.86 | 1:72.79 | 2:69.96 | 3:68.05 | 4:68.05 | 5:67.96 | 6:68.02 | 7:71.10 | 8:71.31 | 9:71.94 | 10:69.29 |
| 70.93 | 1:69.96 | 2:68.05 | 3:68.05 | 4:67.96 | 5:68.02 | 6:71.10 | 7:71.31 | 8:71.94 | 9:69.29 | 10:68.86 |
| 72.51 | 1:68.05 | 2:68.05 | 3:67.96 | 4:68.02 | 5:71.10 | 6:71.31 | 7:71.94 | 8:69.29 | 9:68.86 | 10:70.93 |
| 72.47 | 1:68.05 | 2:67.96 | 3:68.02 | 4:71.10 | 5:71.31 | 6:71.94 | 7:69.29 | 8:68.86 | 9:70.93 | 10:72.51 |
| 72.04 | 1:67.96 | 2:68.02 | 3:71.10 | 4:71.31 | 5:71.94 | 6:69.29 | 7:68.86 | 8:70.93 | 9:72.51 | 10:72.47 |
| 69.71 | 1:68.02 | 2:71.10 | 3:71.31 | 4:71.94 | 5:69.29 | 6:68.86 | 7:70.93 | 8:72.51 | 9:72.47 | 10:72.04 |
| 71.55 | 1:71.10 | 2:71.31 | 3:71.94 | 4:69.29 | 5:68.86 | 6:70.93 | 7:72.51 | 8:72.47 | 9:72.04 | 10:69.71 |
| 68.97 | 1:71.31 | 2:71.94 | 3:69.29 | 4:68.86 | 5:70.93 | 6:72.51 | 7:72.47 | 8:72.04 | 9:69.71 | 10:71.55 |
| 65.79 | 1:71.94 | 2:69.29 | 3:68.86 | 4:70.93 | 5:72.51 | 6:72.47 | 7:72.04 | 8:69.71 | 9:71.55 | 10:68.97 |
| 66.02 | 1:69.29 | 2:68.86 | 3:70.93 | 4:72.51 | 5:72.47 | 6:72.04 | 7:69.71 | 8:71.55 | 9:68.97 | 10:65.79 |
| 66.84 | 1:68.86 | 2:70.93 | 3:72.51 | 4:72.47 | 5:72.04 | 6:69.71 | 7:71.55 | 8:68.97 | 9:65.79 | 10:66.02 |
| 66.71 | 1:70.93 | 2:72.51 | 3:72.47 | 4:72.04 | 5:69.71 | 6:71.55 | 7:68.97 | 8:65.79 | 9:66.02 | 10:66.84 |
| 70.61 | 1:72.51 | 2:72.47 | 3:72.04 | 4:69.71 | 5:71.55 | 6:68.97 | 7:65.79 | 8:66.02 | 9:66.84 | 10:66.71 |



**Data set prediksi harga minyak (1 Oktober 2009-31 Oktober 2009)**

|       |         |         |         |         |         |         |         |         |         |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 70.82 | 1:72.47 | 2:72.04 | 3:69.71 | 4:71.55 | 5:68.97 | 6:65.79 | 7:66.02 | 8:66.84 | 9:66.71 | 10:70.61 |
| 69.95 | 1:72.04 | 2:69.71 | 3:71.55 | 4:68.97 | 5:65.79 | 6:66.02 | 7:66.84 | 8:66.71 | 9:70.61 | 10:70.82 |
| 70.41 | 1:69.71 | 2:71.55 | 3:68.97 | 4:65.79 | 5:66.02 | 6:66.84 | 7:66.71 | 8:70.61 | 9:70.82 | 10:69.95 |
| 70.88 | 1:71.55 | 2:68.97 | 3:65.79 | 4:66.02 | 5:66.84 | 6:66.71 | 7:70.61 | 8:70.82 | 9:69.95 | 10:70.41 |
| 69.57 | 1:68.97 | 2:65.79 | 3:66.02 | 4:66.84 | 5:66.71 | 6:70.61 | 7:70.82 | 8:69.95 | 9:70.41 | 10:70.88 |
| 71.69 | 1:65.79 | 2:66.02 | 3:66.84 | 4:66.71 | 5:70.61 | 6:70.82 | 7:69.95 | 8:70.41 | 9:70.88 | 10:69.57 |
| 71.77 | 1:66.02 | 2:66.84 | 3:66.71 | 4:70.61 | 5:70.82 | 6:69.95 | 7:70.41 | 8:70.88 | 9:69.57 | 10:71.69 |
| 73.27 | 1:66.84 | 2:66.71 | 3:70.61 | 4:70.82 | 5:69.95 | 6:70.41 | 7:70.88 | 8:69.57 | 9:71.69 | 10:71.77 |
| 74.15 | 1:66.71 | 2:70.61 | 3:70.82 | 4:69.95 | 5:70.41 | 6:70.88 | 7:69.57 | 8:71.69 | 9:71.77 | 10:73.27 |
| 75.18 | 1:70.61 | 2:70.82 | 3:69.95 | 4:70.41 | 5:70.88 | 6:69.57 | 7:71.69 | 8:71.77 | 9:73.27 | 10:74.15 |
| 77.58 | 1:70.82 | 2:69.95 | 3:70.41 | 4:70.88 | 5:69.57 | 6:71.69 | 7:71.77 | 8:73.27 | 9:74.15 | 10:75.18 |
| 78.53 | 1:69.95 | 2:70.41 | 3:70.88 | 4:69.57 | 5:71.69 | 6:71.77 | 7:73.27 | 8:74.15 | 9:75.18 | 10:77.58 |
| 79.61 | 1:70.41 | 2:70.88 | 3:69.57 | 4:71.69 | 5:71.77 | 6:73.27 | 7:74.15 | 8:75.18 | 9:77.58 | 10:78.53 |
| 79.09 | 1:70.88 | 2:69.57 | 3:71.69 | 4:71.77 | 5:73.27 | 6:74.15 | 7:75.18 | 8:77.58 | 9:78.53 | 10:79.61 |
| 81.37 | 1:69.57 | 2:71.69 | 3:71.77 | 4:73.27 | 5:74.15 | 6:75.18 | 7:77.58 | 8:78.53 | 9:79.61 | 10:79.09 |
| 81.19 | 1:71.69 | 2:71.77 | 3:73.27 | 4:74.15 | 5:75.18 | 6:77.58 | 7:78.53 | 8:79.61 | 9:79.09 | 10:81.37 |
| 80.50 | 1:71.77 | 2:73.27 | 3:74.15 | 4:75.18 | 5:77.58 | 6:78.53 | 7:79.61 | 8:79.09 | 9:81.37 | 10:81.19 |
| 78.68 | 1:73.27 | 2:74.15 | 3:75.18 | 4:77.58 | 5:78.53 | 6:79.61 | 7:79.09 | 8:81.37 | 9:81.19 | 10:80.50 |
| 79.55 | 1:74.15 | 2:75.18 | 3:77.58 | 4:78.53 | 5:79.61 | 6:79.09 | 7:81.37 | 8:81.19 | 9:80.50 | 10:78.68 |
| 77.46 | 1:75.18 | 2:77.58 | 3:78.53 | 4:79.61 | 5:79.09 | 6:81.37 | 7:81.19 | 8:80.50 | 9:78.68 | 10:79.55 |
| 79.87 | 1:77.58 | 2:78.53 | 3:79.61 | 4:79.09 | 5:81.37 | 6:81.19 | 7:80.50 | 8:78.68 | 9:79.55 | 10:77.46 |

## LAMPIRAN II

### DATA SET NILAI TUKAR RUPIAH

#### Data set pelatihan nilai tukar Rupiah (1 Januari 2009-30 September 2009)

```

11098.80 1:11210.80 2:11236.00 3:11210.80 4:11111.10 5:11098.80 6:11173.20 7:11148.30
8:10976.90 9:10976.90 10:11111.10
11098.80 1:11236.00 2:11210.80 3:11111.10 4:11098.80 5:11173.20 6:11148.30 7:10976.90
8:10976.90 9:11111.10 10:11098.80
11098.80 1:11210.80 2:11111.10 3:11098.80 4:11173.20 5:11148.30 6:10976.90 7:10976.90
8:11111.10 9:11098.80 10:11098.80
11236.00 1:11111.10 2:11098.80 3:11173.20 4:11148.30 5:10976.90 6:10976.90 7:11111.10
8:11098.80 9:11098.80 10:11098.80
11160.70 1:11098.80 2:11173.20 3:11148.30 4:10976.90 5:10976.90 6:11111.10 7:11098.80
8:11098.80 9:11098.80 10:11236.00
11223.30 1:11173.20 2:11148.30 3:10976.90 4:10976.90 5:11111.10 6:11098.80 7:11098.80
8:11098.80 9:11236.00 10:11160.70
11074.20 1:11148.30 2:10976.90 3:10976.90 4:11111.10 5:11098.80 6:11098.80 7:11098.80
8:11236.00 9:11160.70 10:11223.30
11037.50 1:10976.90 2:10976.90 3:11111.10 4:11098.80 5:11098.80 6:11098.80 7:11236.00
8:11160.70 9:11223.30 10:11074.20
11025.40 1:10976.90 2:11111.10 3:11098.80 4:11098.80 5:11098.80 6:11236.00 7:11160.70
8:11223.30 9:11074.20 10:11037.50
11098.80 1:11111.10 2:11098.80 3:11098.80 4:11098.80 5:11236.00 6:11160.70 7:11223.30
8:11074.20 9:11037.50 10:11025.40
11337.90 1:11098.80 2:11098.80 3:11098.80 4:11236.00 5:11160.70 6:11223.30 7:11074.20
8:11037.50 9:11025.40 10:11098.80
11325.00 1:11098.80 2:11098.80 3:11236.00 4:11160.70 5:11223.30 6:11074.20 7:11037.50
8:11025.40 9:11098.80 10:11337.90
11223.30 1:11098.80 2:11236.00 3:11160.70 4:11223.30 5:11074.20 6:11037.50 7:11025.40
8:11098.80 9:11337.90 10:11325.00
11363.60 1:11236.00 2:11160.70 3:11223.30 4:11074.20 5:11037.50 6:11025.40 7:11098.80
8:11337.90 9:11325.00 10:11223.30
11325.00 1:11160.70 2:11223.30 3:11074.20 4:11037.50 5:11025.40 6:11098.80 7:11337.90
8:11325.00 9:11223.30 10:11363.60
11325.00 1:11223.30 2:11074.20 3:11037.50 4:11025.40 5:11098.80 6:11337.90 7:11325.00
8:11223.30 9:11363.60 10:11325.00
11261.30 1:11074.20 2:11037.50 3:11025.40 4:11098.80 5:11337.90 6:11325.00 7:11223.30
8:11363.60 9:11325.00 10:11325.00
11123.50 1:11037.50 2:11025.40 3:11098.80 4:11337.90 5:11325.00 6:11223.30 7:11363.60
8:11325.00 9:11325.00 10:11261.30
11185.70 1:11025.40 2:11098.80 3:11337.90 4:11325.00 5:11223.30 6:11363.60 7:11325.00
8:11325.00 9:11261.30 10:11123.50
11467.90 1:11098.80 2:11337.90 3:11325.00 4:11223.30 5:11363.60 6:11325.00 7:11325.00
8:11261.30 9:11123.50 10:11185.70
11507.50 1:11337.90 2:11325.00 3:11223.30 4:11363.60 5:11325.00 6:11325.00 7:11261.30
8:11123.50 9:11185.70 10:11467.90
11415.50 1:11325.00 2:11223.30 3:11363.60 4:11325.00 5:11325.00 6:11261.30 7:11123.50
8:11185.70 9:11467.90 10:11507.50
11415.50 1:11223.30 2:11363.60 3:11325.00 4:11325.00 5:11261.30 6:11123.50 7:11185.70
8:11467.90 9:11507.50 10:11415.50
11507.50 1:11363.60 2:11325.00 3:11325.00 4:11261.30 5:11123.50 6:11185.70 7:11467.90
8:11507.50 9:11415.50 10:11415.50
11737.10 1:11325.00 2:11325.00 3:11261.30 4:11123.50 5:11185.70 6:11467.90 7:11507.50
8:11415.50 9:11415.50 10:11507.50
11792.50 1:11325.00 2:11261.30 3:11123.50 4:11185.70 5:11467.90 6:11507.50 7:11415.50
8:11415.50 9:11507.50 10:11737.10
11723.30 1:11261.30 2:11123.50 3:11185.70 4:11467.90 5:11507.50 6:11415.50 7:11415.50

```

8:11507.50 9:11737.10 10:11792.50  
 11709.60 1:11123.50 2:11185.70 3:11467.90 4:11507.50 5:11415.50 6:11415.50 7:11507.50  
 8:11737.10 9:11792.50 10:11723.30  
 11641.40 1:11185.70 2:11467.90 3:11507.50 4:11415.50 5:11415.50 6:11507.50 7:11737.10  
 8:11792.50 9:11723.30 10:11709.60  
 11655.00 1:11467.90 2:11507.50 3:11415.50 4:11415.50 5:11507.50 6:11737.10 7:11792.50  
 8:11723.30 9:11709.60 10:11641.40  
 11668.60 1:11507.50 2:11415.50 3:11415.50 4:11507.50 5:11737.10 6:11792.50 7:11723.30  
 8:11709.60 9:11641.40 10:11655.00  
 11904.80 1:11415.50 2:11415.50 3:11507.50 4:11737.10 5:11792.50 6:11723.30 7:11709.60  
 8:11641.40 9:11655.00 10:11668.60  
 11904.80 1:11415.50 2:11507.50 3:11737.10 4:11792.50 5:11723.30 6:11709.60 7:11641.40  
 8:11655.00 9:11668.60 10:11904.80  
 11919.00 1:11507.50 2:11737.10 3:11792.50 4:11723.30 5:11709.60 6:11641.40 7:11655.00  
 8:11668.60 9:11904.80 10:11904.80  
 11806.40 1:11737.10 2:11792.50 3:11723.30 4:11709.60 5:11641.40 6:11655.00 7:11668.60  
 8:11904.80 9:11904.80 10:11919.00  
 11862.40 1:11792.50 2:11723.30 3:11709.60 4:11641.40 5:11655.00 6:11668.60 7:11904.80  
 8:11904.80 9:11919.00 10:11806.40  
 11876.50 1:11723.30 2:11709.60 3:11641.40 4:11655.00 5:11668.60 6:11904.80 7:11904.80  
 8:11919.00 9:11806.40 10:11862.40  
 11976.00 1:11709.60 2:11641.40 3:11655.00 4:11668.60 5:11904.80 6:11904.80 7:11919.00  
 8:11806.40 9:11862.40 10:11876.50  
 12091.90 1:11641.40 2:11655.00 3:11668.60 4:11904.80 5:11904.80 6:11919.00 7:11806.40  
 8:11862.40 9:11876.50 10:11976.00  
 12048.20 1:11655.00 2:11668.60 3:11904.80 4:11904.80 5:11919.00 6:11806.40 7:11862.40  
 8:11876.50 9:11976.00 10:12091.90  
 11890.60 1:11668.60 2:11904.80 3:11904.80 4:11919.00 5:11806.40 6:11862.40 7:11876.50  
 8:11976.00 9:12091.90 10:12048.20  
 11976.00 1:11904.80 2:11904.80 3:11919.00 4:11806.40 5:11862.40 6:11876.50 7:11976.00  
 8:12091.90 9:12048.20 10:11890.60  
 11778.60 1:11904.80 2:11919.00 3:11806.40 4:11862.40 5:11876.50 6:11976.00 7:12091.90  
 8:12048.20 9:11890.60 10:11976.00  
 11792.50 1:11919.00 2:11806.40 3:11862.40 4:11876.50 5:11976.00 6:12091.90 7:12048.20  
 8:11890.60 9:11976.00 10:11778.60  
 11848.30 1:11806.40 2:11862.40 3:11876.50 4:11976.00 5:12091.90 6:12048.20 7:11890.60  
 8:11976.00 9:11778.60 10:11792.50  
 11961.70 1:11862.40 2:11876.50 3:11976.00 4:12091.90 5:12048.20 6:11890.60 7:11976.00  
 8:11778.60 9:11792.50 10:11848.30  
 11919.00 1:11876.50 2:11976.00 3:12091.90 4:12048.20 5:11890.60 6:11976.00 7:11778.60  
 8:11792.50 9:11848.30 10:11961.70  
 11976.00 1:11976.00 2:12091.90 3:12048.20 4:11890.60 5:11976.00 6:11778.60 7:11792.50  
 8:11848.30 9:11961.70 10:11919.00  
 12106.50 1:12091.90 2:12048.20 3:11890.60 4:11976.00 5:11778.60 6:11792.50 7:11848.30  
 8:11961.70 9:11919.00 10:11976.00  
 12077.30 1:12048.20 2:11890.60 3:11976.00 4:11778.60 5:11792.50 6:11848.30 7:11961.70  
 8:11919.00 9:11976.00 10:12106.50  
 12091.90 1:11890.60 2:11976.00 3:11778.60 4:11792.50 5:11848.30 6:11961.70 7:11919.00  
 8:11976.00 9:12106.50 10:12077.30  
 12165.50 1:11976.00 2:11778.60 3:11792.50 4:11848.30 5:11961.70 6:11919.00 7:11976.00  
 8:12106.50 9:12077.30 10:12091.90  
 12091.90 1:11778.60 2:11792.50 3:11848.30 4:11961.70 5:11919.00 6:11976.00 7:12106.50  
 8:12077.30 9:12091.90 10:12165.50  
 12062.70 1:11792.50 2:11848.30 3:11961.70 4:11919.00 5:11976.00 6:12106.50 7:12077.30  
 8:12091.90 9:12165.50 10:12091.90  
 12180.30 1:11848.30 2:11961.70 3:11919.00 4:11976.00 5:12106.50 6:12077.30 7:12091.90  
 8:12165.50 9:12091.90 10:12062.70  
 12004.80 1:11961.70 2:11919.00 3:11976.00 4:12106.50 5:12077.30 6:12091.90 7:12165.50  
 8:12091.90 9:12062.70 10:12180.30  
 12121.20 1:11919.00 2:11976.00 3:12106.50 4:12077.30 5:12091.90 6:12165.50 7:12091.90  
 8:12062.70 9:12180.30 10:12004.80  
 12106.50 1:11976.00 2:12106.50 3:12077.30 4:12091.90 5:12165.50 6:12091.90 7:12062.70  
 8:12180.30 9:12004.80 10:12121.20

12106.50 1:12106.50 2:12077.30 3:12091.90 4:12165.50 5:12091.90 6:12062.70 7:12180.30  
 8:12004.80 9:12121.20 10:12106.50  
 12033.70 1:12077.30 2:12091.90 3:12165.50 4:12091.90 5:12062.70 6:12180.30 7:12004.80  
 8:12121.20 9:12106.50 10:12106.50  
 11976.00 1:12091.90 2:12165.50 3:12091.90 4:12062.70 5:12180.30 6:12004.80 7:12121.20  
 8:12106.50 9:12106.50 10:12033.70  
 11961.70 1:12165.50 2:12091.90 3:12062.70 4:12180.30 5:12004.80 6:12121.20 7:12106.50  
 8:12106.50 9:12033.70 10:11976.00  
 11947.40 1:12091.90 2:12062.70 3:12180.30 4:12004.80 5:12121.20 6:12106.50 7:12106.50  
 8:12033.70 9:11976.00 10:11961.70  
 11990.40 1:12062.70 2:12180.30 3:12004.80 4:12121.20 5:12106.50 6:12106.50 7:12033.70  
 8:11976.00 9:11961.70 10:11947.40  
 11990.40 1:12180.30 2:12004.80 3:12121.20 4:12106.50 5:12106.50 6:12033.70 7:11976.00  
 8:11961.70 9:11947.40 10:11990.40  
 11976.00 1:12004.80 2:12121.20 3:12106.50 4:12106.50 5:12033.70 6:11976.00 7:11961.70  
 8:11947.40 9:11990.40 10:11990.40  
 12019.20 1:12121.20 2:12106.50 3:12106.50 4:12033.70 5:11976.00 6:11961.70 7:11947.40  
 8:11990.40 9:11990.40 10:11976.00  
 11919.00 1:12106.50 2:12106.50 3:12033.70 4:11976.00 5:11961.70 6:11947.40 7:11990.40  
 8:11990.40 9:11976.00 10:12019.20  
 11778.60 1:12106.50 2:12033.70 3:11976.00 4:11961.70 5:11947.40 6:11990.40 7:11990.40  
 8:11976.00 9:12019.20 10:11919.00  
 12019.20 1:12033.70 2:11976.00 3:11961.70 4:11947.40 5:11990.40 6:11990.40 7:11976.00  
 8:12019.20 9:11919.00 10:11778.60  
 11806.40 1:11976.00 2:11961.70 3:11947.40 4:11990.40 5:11990.40 6:11976.00 7:12019.20  
 8:11919.00 9:11778.60 10:12019.20  
 11806.40 1:11961.70 2:11947.40 3:11990.40 4:11990.40 5:11976.00 6:12019.20 7:11919.00  
 8:11778.60 9:12019.20 10:11806.40  
 11764.70 1:11947.40 2:11990.40 3:11990.40 4:11976.00 5:12019.20 6:11919.00 7:11778.60  
 8:12019.20 9:11806.40 10:11806.40  
 11820.30 1:11990.40 2:11990.40 3:11976.00 4:12019.20 5:11919.00 6:11778.60 7:12019.20  
 8:11806.40 9:11806.40 10:11764.70  
 11792.50 1:11990.40 2:11976.00 3:12019.20 4:11919.00 5:11778.60 6:12019.20 7:11806.40  
 8:11806.40 9:11764.70 10:11820.30  
 11641.40 1:11976.00 2:12019.20 3:11919.00 4:11778.60 5:12019.20 6:11806.40 7:11806.40  
 8:11764.70 9:11820.30 10:11792.50  
 11627.90 1:12019.20 2:11919.00 3:11778.60 4:12019.20 5:11806.40 6:11806.40 7:11764.70  
 8:11820.30 9:11792.50 10:11641.40  
 11534.00 1:11919.00 2:11778.60 3:12019.20 4:11806.40 5:11806.40 6:11764.70 7:11820.30  
 8:11792.50 9:11641.40 10:11627.90  
 11547.30 1:11778.60 2:12019.20 3:11806.40 4:11806.40 5:11764.70 6:11820.30 7:11792.50  
 8:11641.40 9:11627.90 10:11534.00  
 11627.90 1:12019.20 2:11806.40 3:11806.40 4:11764.70 5:11820.30 6:11792.50 7:11641.40  
 8:11627.90 9:11534.00 10:11547.30  
 11574.10 1:11806.40 2:11806.40 3:11764.70 4:11820.30 5:11792.50 6:11641.40 7:11627.90  
 8:11534.00 9:11547.30 10:11627.90  
 11641.40 1:11806.40 2:11764.70 3:11820.30 4:11792.50 5:11641.40 6:11627.90 7:11534.00  
 8:11547.30 9:11627.90 10:11574.10  
 11520.70 1:11764.70 2:11820.30 3:11792.50 4:11641.40 5:11627.90 6:11534.00 7:11547.30  
 8:11627.90 9:11574.10 10:11641.40  
 11520.70 1:11820.30 2:11792.50 3:11641.40 4:11627.90 5:11534.00 6:11547.30 7:11627.90  
 8:11574.10 9:11641.40 10:11520.70  
 11428.60 1:11792.50 2:11641.40 3:11627.90 4:11534.00 5:11547.30 6:11627.90 7:11574.10  
 8:11641.40 9:11520.70 10:11520.70  
 11441.60 1:11641.40 2:11627.90 3:11534.00 4:11547.30 5:11627.90 6:11574.10 7:11641.40  
 8:11520.70 9:11520.70 10:11428.60  
 11441.60 1:11627.90 2:11534.00 3:11547.30 4:11627.90 5:11574.10 6:11641.40 7:11520.70  
 8:11520.70 9:11428.60 10:11441.60  
 11507.50 1:11534.00 2:11547.30 3:11627.90 4:11574.10 5:11641.40 6:11520.70 7:11520.70  
 8:11428.60 9:11441.60 10:11441.60  
 11389.50 1:11547.30 2:11627.90 3:11574.10 4:11641.40 5:11520.70 6:11520.70 7:11428.60  
 8:11441.60 9:11441.60 10:11507.50  
 11363.60 1:11627.90 2:11574.10 3:11641.40 4:11520.70 5:11520.70 6:11428.60 7:11441.60

8:11441.60 9:11507.50 10:11389.50  
 11402.50 1:11574.10 2:11641.40 3:11520.70 4:11520.70 5:11428.60 6:11441.60 7:11441.60  
 8:11507.50 9:11389.50 10:11363.60  
 11363.60 1:11641.40 2:11520.70 3:11520.70 4:11428.60 5:11441.60 6:11441.60 7:11507.50  
 8:11389.50 9:11363.60 10:11402.50  
 11350.70 1:11520.70 2:11520.70 3:11428.60 4:11441.60 5:11441.60 6:11507.50 7:11389.50  
 8:11363.60 9:11402.50 10:11363.60  
 11299.40 1:11520.70 2:11428.60 3:11441.60 4:11441.60 5:11507.50 6:11389.50 7:11363.60  
 8:11402.50 9:11363.60 10:11350.70  
 11173.20 1:11428.60 2:11441.60 3:11441.60 4:11507.50 5:11389.50 6:11363.60 7:11402.50  
 8:11363.60 9:11350.70 10:11299.40  
 10952.90 1:11441.60 2:11441.60 3:11507.50 4:11389.50 5:11363.60 6:11402.50 7:11363.60  
 8:11350.70 9:11299.40 10:11173.20  
 10952.90 1:11441.60 2:11507.50 3:11389.50 4:11363.60 5:11402.50 6:11363.60 7:11350.70  
 8:11299.40 9:11173.20 10:10952.90  
 11111.10 1:11507.50 2:11389.50 3:11363.60 4:11402.50 5:11363.60 6:11350.70 7:11299.40  
 8:11173.20 9:10952.90 10:10952.90  
 11428.60 1:11389.50 2:11363.60 3:11402.50 4:11363.60 5:11350.70 6:11299.40 7:11173.20  
 8:10952.90 9:10952.90 10:11111.10  
 11534.00 1:11363.60 2:11402.50 3:11363.60 4:11350.70 5:11299.40 6:11173.20 7:10952.90  
 8:10952.90 9:11111.10 10:11428.60  
 11049.70 1:11402.50 2:11363.60 3:11350.70 4:11299.40 5:11173.20 6:10952.90 7:10952.90  
 8:11111.10 9:11428.60 10:11534.00  
 10764.30 1:11363.60 2:11350.70 3:11299.40 4:11173.20 5:10952.90 6:10952.90 7:11111.10  
 8:11428.60 9:11534.00 10:11049.70  
 10893.20 1:11350.70 2:11299.40 3:11173.20 4:10952.90 5:10952.90 6:11111.10 7:11428.60  
 8:11534.00 9:11049.70 10:10764.30  
 10869.60 1:11299.40 2:11173.20 3:10952.90 4:10952.90 5:11111.10 6:11428.60 7:11534.00  
 8:11049.70 9:10764.30 10:10893.20  
 10764.30 1:11173.20 2:10952.90 3:10952.90 4:11111.10 5:11428.60 6:11534.00 7:11049.70  
 8:10764.30 9:10893.20 10:10869.60  
 10764.30 1:10952.90 2:10952.90 3:11111.10 4:11428.60 5:11534.00 6:11049.70 7:10764.30  
 8:10893.20 9:10869.60 10:10764.30  
 10752.70 1:10952.90 2:11111.10 3:11428.60 4:11534.00 5:11049.70 6:10764.30 7:10893.20  
 8:10869.60 9:10764.30 10:10764.30  
 10869.60 1:11111.10 2:11428.60 3:11534.00 4:11049.70 5:10764.30 6:10893.20 7:10869.60  
 8:10764.30 9:10764.30 10:10752.70  
 10917.00 1:11428.60 2:11534.00 3:11049.70 4:10764.30 5:10893.20 6:10869.60 7:10764.30  
 8:10764.30 9:10752.70 10:10869.60  
 10752.70 1:11534.00 2:11049.70 3:10764.30 4:10893.20 5:10869.60 6:10764.30 7:10764.30  
 8:10752.70 9:10869.60 10:10917.00  
 10729.60 1:11049.70 2:10764.30 3:10893.20 4:10869.60 5:10764.30 6:10764.30 7:10752.70  
 8:10869.60 9:10917.00 10:10752.70  
 10718.10 1:10764.30 2:10893.20 3:10869.60 4:10764.30 5:10764.30 6:10752.70 7:10869.60  
 8:10917.00 9:10752.70 10:10729.60  
 10661.00 1:10893.20 2:10869.60 3:10764.30 4:10764.30 5:10752.70 6:10869.60 7:10917.00  
 8:10752.70 9:10729.60 10:10718.10  
 10672.40 1:10869.60 2:10764.30 3:10764.30 4:10752.70 5:10869.60 6:10917.00 7:10752.70  
 8:10729.60 9:10718.10 10:10661.00  
 10627.00 1:10764.30 2:10764.30 3:10752.70 4:10869.60 5:10917.00 6:10752.70 7:10729.60  
 8:10718.10 9:10661.00 10:10672.40  
 10526.30 1:10764.30 2:10752.70 3:10869.60 4:10917.00 5:10752.70 6:10729.60 7:10718.10  
 8:10661.00 9:10672.40 10:10627.00  
 10449.30 1:10752.70 2:10869.60 3:10917.00 4:10752.70 5:10729.60 6:10718.10 7:10661.00  
 8:10672.40 9:10627.00 10:10526.30  
 10427.50 1:10869.60 2:10917.00 3:10752.70 4:10729.60 5:10718.10 6:10661.00 7:10672.40  
 8:10627.00 9:10526.30 10:10449.30  
 10460.30 1:10917.00 2:10752.70 3:10729.60 4:10718.10 5:10661.00 6:10672.40 7:10627.00  
 8:10526.30 9:10449.30 10:10427.50  
 10775.90 1:10752.70 2:10729.60 3:10718.10 4:10661.00 5:10672.40 6:10627.00 7:10526.30  
 8:10449.30 9:10427.50 10:10460.30  
 10752.70 1:10729.60 2:10718.10 3:10661.00 4:10672.40 5:10627.00 6:10526.30 7:10449.30  
 8:10427.50 9:10460.30 10:10775.90

10615.70 1:10718.10 2:10661.00 3:10672.40 4:10627.00 5:10526.30 6:10449.30 7:10427.50  
 8:10460.30 9:10775.90 10:10752.70  
 10515.20 1:10661.00 2:10672.40 3:10627.00 4:10526.30 5:10449.30 6:10427.50 7:10460.30  
 8:10775.90 9:10752.70 10:10615.70  
 10395.00 1:10672.40 2:10627.00 3:10526.30 4:10449.30 5:10427.50 6:10460.30 7:10775.90  
 8:10752.70 9:10615.70 10:10515.20  
 10438.40 1:10627.00 2:10526.30 3:10449.30 4:10427.50 5:10460.30 6:10775.90 7:10752.70  
 8:10615.70 9:10515.20 10:10395.00  
 10449.30 1:10526.30 2:10449.30 3:10427.50 4:10460.30 5:10775.90 6:10752.70 7:10615.70  
 8:10515.20 9:10395.00 10:10438.40  
 10395.00 1:10449.30 2:10427.50 3:10460.30 4:10775.90 5:10752.70 6:10615.70 7:10515.20  
 8:10395.00 9:10438.40 10:10449.30  
 10405.80 1:10427.50 2:10460.30 3:10775.90 4:10752.70 5:10615.70 6:10515.20 7:10395.00  
 8:10438.40 9:10449.30 10:10395.00  
 10405.80 1:10460.30 2:10775.90 3:10752.70 4:10615.70 5:10515.20 6:10395.00 7:10438.40  
 8:10449.30 9:10395.00 10:10405.80  
 10309.30 1:10775.90 2:10752.70 3:10615.70 4:10515.20 5:10395.00 6:10438.40 7:10449.30  
 8:10395.00 9:10405.80 10:10405.80  
 10352.00 1:10752.70 2:10615.70 3:10515.20 4:10395.00 5:10438.40 6:10449.30 7:10395.00  
 8:10405.80 9:10405.80 10:10309.30  
 10373.40 1:10615.70 2:10515.20 3:10395.00 4:10438.40 5:10449.30 6:10395.00 7:10405.80  
 8:10405.80 9:10309.30 10:10352.00  
 10245.90 1:10515.20 2:10395.00 3:10438.40 4:10449.30 5:10395.00 6:10405.80 7:10405.80  
 8:10309.30 9:10352.00 10:10373.40  
 10224.90 1:10395.00 2:10438.40 3:10449.30 4:10395.00 5:10405.80 6:10405.80 7:10309.30  
 8:10352.00 9:10373.40 10:10245.90  
 10214.50 1:10438.40 2:10449.30 3:10395.00 4:10405.80 5:10405.80 6:10309.30 7:10352.00  
 8:10373.40 9:10245.90 10:10224.90  
 10224.90 1:10449.30 2:10395.00 3:10405.80 4:10405.80 5:10309.30 6:10352.00 7:10373.40  
 8:10245.90 9:10224.90 10:10214.50  
 10298.70 1:10395.00 2:10405.80 3:10405.80 4:10309.30 5:10352.00 6:10373.40 7:10245.90  
 8:10224.90 9:10214.50 10:10224.90  
 10493.20 1:10405.80 2:10405.80 3:10309.30 4:10352.00 5:10373.40 6:10245.90 7:10224.90  
 8:10214.50 9:10224.90 10:10298.70  
 10615.70 1:10405.80 2:10309.30 3:10352.00 4:10373.40 5:10245.90 6:10224.90 7:10214.50  
 8:10224.90 9:10298.70 10:10493.20  
 10277.50 1:10309.30 2:10352.00 3:10373.40 4:10245.90 5:10224.90 6:10214.50 7:10224.90  
 8:10298.70 9:10493.20 10:10615.70  
 10235.40 1:10352.00 2:10373.40 3:10245.90 4:10224.90 5:10214.50 6:10224.90 7:10298.70  
 8:10493.20 9:10615.70 10:10277.50  
 10235.40 1:10373.40 2:10245.90 3:10224.90 4:10214.50 5:10224.90 6:10298.70 7:10493.20  
 8:10615.70 9:10277.50 10:10235.40  
 10193.70 1:10245.90 2:10224.90 3:10214.50 4:10224.90 5:10298.70 6:10493.20 7:10615.70  
 8:10277.50 9:10235.40 10:10235.40  
 10214.50 1:10224.90 2:10214.50 3:10224.90 4:10298.70 5:10493.20 6:10615.70 7:10277.50  
 8:10235.40 9:10235.40 10:10193.70  
 10288.10 1:10214.50 2:10224.90 3:10298.70 4:10493.20 5:10615.70 6:10277.50 7:10235.40  
 8:10235.40 9:10193.70 10:10214.50  
 10142.00 1:10224.90 2:10298.70 3:10493.20 4:10615.70 5:10277.50 6:10235.40 7:10235.40  
 8:10193.70 9:10214.50 10:10288.10  
 10101.00 1:10298.70 2:10493.20 3:10615.70 4:10277.50 5:10235.40 6:10235.40 7:10193.70  
 8:10214.50 9:10288.10 10:10142.00  
 9990.01 1:10493.20 2:10615.70 3:10277.50 4:10235.40 5:10235.40 6:10193.70 7:10214.50  
 8:10288.10 9:10142.00 10:10101.00  
 9871.67 1:10615.70 2:10277.50 3:10235.40 4:10235.40 5:10193.70 6:10214.50 7:10288.10  
 8:10142.00 9:10101.00 10:9990.01  
 10060.40 1:10277.50 2:10235.40 3:10235.40 4:10193.70 5:10214.50 6:10288.10 7:10142.00  
 8:10101.00 9:9990.01 10:9871.67  
 10050.30 1:10235.40 2:10235.40 3:10193.70 4:10214.50 5:10288.10 6:10142.00 7:10101.00  
 8:9990.01 9:9871.67 10:10060.40  
 10080.60 1:10235.40 2:10193.70 3:10214.50 4:10288.10 5:10142.00 6:10101.00 7:9990.01  
 8:9871.67 9:10060.40 10:10050.30  
 10050.30 1:10193.70 2:10214.50 3:10288.10 4:10142.00 5:10101.00 6:9990.01 7:9871.67

8:10060.40 9:10050.30 10:10080.60  
 10101.00 1:10214.50 2:10288.10 3:10142.00 4:10101.00 5:9990.01 6:9871.67 7:10060.40  
 8:10050.30 9:10080.60 10:10050.30  
 10121.50 1:10288.10 2:10142.00 3:10101.00 4:9990.01 5:9871.67 6:10060.40 7:10050.30  
 8:10080.60 9:10050.30 10:10101.00  
 10121.50 1:10142.00 2:10101.00 3:9990.01 4:9871.67 5:10060.40 6:10050.30 7:10080.60  
 8:10050.30 9:10101.00 10:10121.50  
 10214.50 1:10101.00 2:9990.01 3:9871.67 4:10060.40 5:10050.30 6:10080.60 7:10050.30  
 8:10101.00 9:10121.50 10:10121.50  
 10172.90 1:9990.01 2:9871.67 3:10060.40 4:10050.30 5:10080.60 6:10050.30 7:10101.00  
 8:10121.50 9:10121.50 10:10214.50  
 10204.10 1:9871.67 2:10060.40 3:10050.30 4:10080.60 5:10050.30 6:10101.00 7:10121.50  
 8:10121.50 9:10214.50 10:10172.90  
 10288.10 1:10060.40 2:10050.30 3:10080.60 4:10050.30 5:10101.00 6:10121.50 7:10121.50  
 8:10214.50 9:10172.90 10:10204.10  
 10471.20 1:10050.30 2:10080.60 3:10050.30 4:10101.00 5:10121.50 6:10121.50 7:10214.50  
 8:10172.90 9:10204.10 10:10288.10  
 10449.30 1:10080.60 2:10050.30 3:10101.00 4:10121.50 5:10121.50 6:10214.50 7:10172.90  
 8:10204.10 9:10288.10 10:10471.20  
 10471.20 1:10050.30 2:10101.00 3:10121.50 4:10121.50 5:10214.50 6:10172.90 7:10204.10  
 8:10288.10 9:10471.20 10:10449.30  
 10427.50 1:10101.00 2:10121.50 3:10121.50 4:10214.50 5:10172.90 6:10204.10 7:10288.10  
 8:10471.20 9:10449.30 10:10471.20  
 10224.90 1:10121.50 2:10121.50 3:10214.50 4:10172.90 5:10204.10 6:10288.10 7:10471.20  
 8:10449.30 9:10471.20 10:10427.50  
 10373.40 1:10121.50 2:10214.50 3:10172.90 4:10204.10 5:10288.10 6:10471.20 7:10449.30  
 8:10471.20 9:10427.50 10:10224.90  
 10352.00 1:10214.50 2:10172.90 3:10204.10 4:10288.10 5:10471.20 6:10449.30 7:10471.20  
 8:10427.50 9:10224.90 10:10373.40  
 10277.50 1:10172.90 2:10204.10 3:10288.10 4:10471.20 5:10449.30 6:10471.20 7:10427.50  
 8:10224.90 9:10373.40 10:10352.00  
 10298.70 1:10204.10 2:10288.10 3:10471.20 4:10449.30 5:10471.20 6:10427.50 7:10224.90  
 8:10373.40 9:10352.00 10:10277.50  
 10298.70 1:10288.10 2:10471.20 3:10449.30 4:10471.20 5:10427.50 6:10224.90 7:10373.40  
 8:10352.00 9:10277.50 10:10298.70  
 10298.70 1:10471.20 2:10449.30 3:10471.20 4:10427.50 5:10224.90 6:10373.40 7:10352.00  
 8:10277.50 9:10298.70 10:10298.70  
 10245.90 1:10449.30 2:10471.20 3:10427.50 4:10224.90 5:10373.40 6:10352.00 7:10277.50  
 8:10298.70 9:10298.70 10:10298.70  
 10162.60 1:10471.20 2:10427.50 3:10224.90 4:10373.40 5:10352.00 6:10277.50 7:10298.70  
 8:10298.70 9:10298.70 10:10245.90  
 10204.10 1:10427.50 2:10224.90 3:10373.40 4:10352.00 5:10277.50 6:10298.70 7:10298.70  
 8:10298.70 9:10245.90 10:10162.60  
 10245.90 1:10224.90 2:10373.40 3:10352.00 4:10277.50 5:10298.70 6:10298.70 7:10298.70  
 8:10245.90 9:10162.60 10:10204.10  
 10256.40 1:10373.40 2:10352.00 3:10277.50 4:10298.70 5:10298.70 6:10298.70 7:10245.90  
 8:10162.60 9:10204.10 10:10245.90  
 10245.90 1:10352.00 2:10277.50 3:10298.70 4:10298.70 5:10298.70 6:10245.90 7:10162.60  
 8:10204.10 9:10245.90 10:10256.40  
 10288.10 1:10277.50 2:10298.70 3:10298.70 4:10298.70 5:10245.90 6:10162.60 7:10204.10  
 8:10245.90 9:10256.40 10:10245.90  
 10266.90 1:10298.70 2:10298.70 3:10298.70 4:10245.90 5:10162.60 6:10204.10 7:10245.90  
 8:10256.40 9:10245.90 10:10288.10  
 10319.90 1:10298.70 2:10298.70 3:10245.90 4:10162.60 5:10204.10 6:10245.90 7:10256.40  
 8:10245.90 9:10288.10 10:10266.90  
 10256.40 1:10298.70 2:10245.90 3:10162.60 4:10204.10 5:10245.90 6:10256.40 7:10245.90  
 8:10288.10 9:10266.90 10:10319.90  
 10309.30 1:10245.90 2:10162.60 3:10204.10 4:10245.90 5:10256.40 6:10245.90 7:10288.10  
 8:10266.90 9:10319.90 10:10256.40  
 10145.00 1:10162.60 2:10204.10 3:10245.90 4:10256.40 5:10245.90 6:10288.10 7:10266.90  
 8:10319.90 9:10256.40 10:10309.30  
 10190.00 1:10204.10 2:10245.90 3:10256.40 4:10245.90 5:10288.10 6:10266.90 7:10319.90  
 8:10256.40 9:10309.30 10:10145.00

10185.00 1:10245.90 2:10256.40 3:10245.90 4:10288.10 5:10266.90 6:10319.90 7:10256.40  
 8:10309.30 9:10145.00 10:10190.00  
 10115.00 1:10256.40 2:10245.90 3:10288.10 4:10266.90 5:10319.90 6:10256.40 7:10309.30  
 8:10145.00 9:10190.00 10:10185.00  
 10115.00 1:10245.90 2:10288.10 3:10266.90 4:10319.90 5:10256.40 6:10309.30 7:10145.00  
 8:10190.00 9:10185.00 10:10115.00  
 10115.00 1:10288.10 2:10266.90 3:10319.90 4:10256.40 5:10309.30 6:10145.00 7:10190.00  
 8:10185.00 9:10115.00 10:10115.00  
 10115.00 1:10266.90 2:10319.90 3:10256.40 4:10309.30 5:10145.00 6:10190.00 7:10185.00  
 8:10115.00 9:10115.00 10:10115.00  
 10115.00 1:10319.90 2:10256.40 3:10309.30 4:10145.00 5:10190.00 6:10185.00 7:10115.00  
 8:10115.00 9:10115.00 10:10115.00  
 10115.00 1:10256.40 2:10309.30 3:10145.00 4:10190.00 5:10185.00 6:10115.00 7:10115.00  
 8:10115.00 9:10115.00 10:10115.00  
 10172.90 1:10309.30 2:10145.00 3:10190.00 4:10185.00 5:10115.00 6:10115.00 7:10115.00  
 8:10115.00 9:10115.00 10:10115.00  
 10060.40 1:10145.00 2:10190.00 3:10185.00 4:10115.00 5:10115.00 6:10115.00 7:10115.00  
 8:10115.00 9:10115.00 10:10172.90  
 10030.10 1:10190.00 2:10185.00 3:10115.00 4:10115.00 5:10115.00 6:10115.00 7:10115.00  
 8:10115.00 9:10172.90 10:10060.40  
 10030.10 1:10185.00 2:10115.00 3:10115.00 4:10115.00 5:10115.00 6:10115.00 7:10115.00  
 8:10172.90 9:10060.40 10:10030.10  
 10060.40 1:10115.00 2:10115.00 3:10115.00 4:10115.00 5:10115.00 6:10115.00 7:10172.90  
 8:10060.40 9:10030.10 10:10030.10  
 10060.40 1:10115.00 2:10115.00 3:10115.00 4:10115.00 5:10115.00 6:10172.90 7:10060.40  
 8:10030.10 9:10030.10 10:10060.40  
 10060.40 1:10115.00 2:10115.00 3:10115.00 4:10115.00 5:10172.90 6:10060.40 7:10030.10  
 8:10030.10 9:10060.40 10:10060.40  
 10030.10 1:10115.00 2:10115.00 3:10115.00 4:10172.90 5:10060.40 6:10030.10 7:10030.10  
 8:10060.40 9:10060.40 10:10060.40  
 10020.00 1:10115.00 2:10115.00 3:10172.90 4:10060.40 5:10030.10 6:10030.10 7:10060.40  
 8:10060.40 9:10060.40 10:10030.10  
 10040.20 1:10115.00 2:10172.90 3:10060.40 4:10030.10 5:10030.10 6:10060.40 7:10060.40  
 8:10060.40 9:10030.10 10:10020.00  
 9970.09 1:10172.90 2:10060.40 3:10030.10 4:10030.10 5:10060.40 6:10060.40 7:10060.40  
 8:10030.10 9:10020.00 10:10040.20  
 9809.20 1:10060.40 2:10030.10 3:10030.10 4:10060.40 5:10060.40 6:10060.40 7:10030.10  
 8:10020.00 9:10040.20 10:9970.09  
 9813.54 1:10030.10 2:10030.10 3:10060.40 4:10060.40 5:10060.40 6:10030.10 7:10020.00  
 8:10040.20 9:9970.09 10:9809.20  
 9813.54 1:10030.10 2:10060.40 3:10060.40 4:10060.40 5:10030.10 6:10020.00 7:10040.20  
 8:9970.09 9:9809.20 10:9813.54  
 9756.10 1:10060.40 2:10060.40 3:10060.40 4:10030.10 5:10020.00 6:10040.20 7:9970.09  
 8:9809.20 9:9813.54 10:9813.54  
 9708.74 1:10060.40 2:10060.40 3:10030.10 4:10020.00 5:10040.20 6:9970.09 7:9809.20  
 8:9813.54 9:9813.54 10:9756.10  
 9756.10 1:10060.40 2:10030.10 3:10020.00 4:10040.20 5:9970.09 6:9809.20 7:9813.54  
 8:9813.54 9:9756.10 10:9708.74  
 9910.80 1:10030.10 2:10020.00 3:10040.20 4:9970.09 5:9809.20 6:9813.54 7:9813.54  
 8:9756.10 9:9708.74 10:9756.10  
 9950.25 1:10020.00 2:10040.20 3:9970.09 4:9809.20 5:9813.54 6:9813.54 7:9756.10  
 8:9708.74 9:9756.10 10:9910.80  
 9970.09 1:10040.20 2:9970.09 3:9809.20 4:9813.54 5:9813.54 6:9756.10 7:9708.74  
 8:9756.10 9:9910.80 10:9950.25  
 9980.04 1:9970.09 2:9809.20 3:9813.54 4:9813.54 5:9756.10 6:9708.74 7:9756.10  
 8:9910.80 9:9950.25 10:9970.09  
 9960.16 1:9809.20 2:9813.54 3:9813.54 4:9756.10 5:9708.74 6:9756.10 7:9910.80  
 8:9950.25 9:9970.09 10:9980.04  
 9940.36 1:9813.54 2:9813.54 3:9756.10 4:9708.74 5:9756.10 6:9910.80 7:9950.25  
 8:9970.09 9:9980.04 10:9960.16  
 9950.25 1:9813.54 2:9756.10 3:9708.74 4:9756.10 5:9910.80 6:9950.25 7:9970.09  
 8:9980.04 9:9960.16 10:9940.36  
 9940.36 1:9756.10 2:9708.74 3:9756.10 4:9910.80 5:9950.25 6:9970.09 7:9980.04



8:9960.16 9:9940.36 10:9950.25  
 9970.09 1:9708.74 2:9756.10 3:9910.80 4:9950.25 5:9970.09 6:9980.04 7:9960.16  
 8:9940.36 9:9950.25 10:9940.36  
 9980.04 1:9756.10 2:9910.80 3:9950.25 4:9970.09 5:9980.04 6:9960.16 7:9940.36  
 8:9950.25 9:9940.36 10:9970.09  
 9980.04 1:9910.80 2:9950.25 3:9970.09 4:9980.04 5:9960.16 6:9940.36 7:9950.25  
 8:9940.36 9:9970.09 10:9980.04  
 10060.40 1:9950.25 2:9970.09 3:9980.04 4:9960.16 5:9940.36 6:9950.25 7:9940.36  
 8:9970.09 9:9980.04 10:9980.04  
 10050.30 1:9970.09 2:9980.04 3:9960.16 4:9940.36 5:9950.25 6:9940.36 7:9970.09  
 8:9980.04 9:9980.04 10:10060.40  
 10050.30 1:9980.04 2:9960.16 3:9940.36 4:9950.25 5:9940.36 6:9970.09 7:9980.04  
 8:9980.04 9:10060.40 10:10050.30  
 10020.00 1:9960.16 2:9940.36 3:9950.25 4:9940.36 5:9970.09 6:9980.04 7:9980.04  
 8:10060.40 9:10050.30 10:10050.30  
 10050.30 1:9940.36 2:9950.25 3:9940.36 4:9970.09 5:9980.04 6:9980.04 7:10060.40  
 8:10050.30 9:10050.30 10:10020.00  
 10020.00 1:9950.25 2:9940.36 3:9970.09 4:9980.04 5:9980.04 6:10060.40 7:10050.30  
 8:10050.30 9:10020.00 10:10050.30  
 10020.00 1:9940.36 2:9970.09 3:9980.04 4:9980.04 5:10060.40 6:10050.30 7:10050.30  
 8:10020.00 9:10050.30 10:10020.00  
 10020.00 1:9970.09 2:9980.04 3:9980.04 4:10060.40 5:10050.30 6:10050.30 7:10020.00  
 8:10050.30 9:10020.00 10:10020.00  
 10020.00 1:9980.04 2:9980.04 3:10060.40 4:10050.30 5:10050.30 6:10020.00 7:10050.30  
 8:10020.00 9:10020.00 10:10020.00  
 10050.30 1:9980.04 2:10060.40 3:10050.30 4:10050.30 5:10020.00 6:10050.30 7:10020.00  
 8:10020.00 9:10020.00 10:10020.00  
 10070.50 1:10060.40 2:10050.30 3:10050.30 4:10020.00 5:10050.30 6:10020.00 7:10020.00  
 8:10020.00 9:10020.00 10:10050.30  
 10070.50 1:10050.30 2:10050.30 3:10020.00 4:10050.30 5:10020.00 6:10020.00 7:10020.00  
 8:10020.00 9:10050.30 10:10070.50  
 10050.30 1:10050.30 2:10020.00 3:10050.30 4:10020.00 5:10020.00 6:10020.00 7:10020.00  
 8:10050.30 9:10070.50 10:10070.50  
 10050.30 1:10020.00 2:10050.30 3:10020.00 4:10020.00 5:10020.00 6:10020.00 7:10050.30  
 8:10070.50 9:10070.50 10:10050.30  
 10090.80 1:10050.30 2:10020.00 3:10020.00 4:10020.00 5:10020.00 6:10050.30 7:10070.50  
 8:10070.50 9:10050.30 10:10050.30  
 10121.50 1:10020.00 2:10020.00 3:10020.00 4:10020.00 5:10050.30 6:10070.50 7:10070.50  
 8:10050.30 9:10050.30 10:10090.80  
 10131.70 1:10020.00 2:10020.00 3:10020.00 4:10050.30 5:10070.50 6:10070.50 7:10050.30  
 8:10050.30 9:10090.80 10:10121.50  
 10111.20 1:10020.00 2:10020.00 3:10050.30 4:10070.50 5:10070.50 6:10050.30 7:10050.30  
 8:10090.80 9:10121.50 10:10131.70  
 10131.70 1:10020.00 2:10050.30 3:10070.50 4:10070.50 5:10050.30 6:10050.30 7:10090.80  
 8:10121.50 9:10131.70 10:10111.20  
 10142.00 1:10050.30 2:10070.50 3:10070.50 4:10050.30 5:10050.30 6:10090.80 7:10121.50  
 8:10131.70 9:10111.20 10:10131.70  
 10131.70 1:10070.50 2:10070.50 3:10050.30 4:10050.30 5:10090.80 6:10121.50 7:10131.70  
 8:10111.20 9:10131.70 10:10142.00  
 10101.00 1:10070.50 2:10050.30 3:10050.30 4:10090.80 5:10121.50 6:10131.70 7:10111.20  
 8:10131.70 9:10142.00 10:10131.70  
 9990.01 1:10050.30 2:10050.30 3:10090.80 4:10121.50 5:10131.70 6:10111.20 7:10131.70  
 8:10142.00 9:10131.70 10:10101.00  
 9900.99 1:10050.30 2:10090.80 3:10121.50 4:10131.70 5:10111.20 6:10131.70 7:10142.00  
 8:10131.70 9:10101.00 10:9990.01  
 9920.63 1:10090.80 2:10121.50 3:10131.70 4:10111.20 5:10131.70 6:10142.00 7:10131.70  
 8:10101.00 9:9990.01 10:9900.99  
 9900.99 1:10121.50 2:10131.70 3:10111.20 4:10131.70 5:10142.00 6:10131.70 7:10101.00  
 8:9990.01 9:9900.99 10:9920.63  
 9910.80 1:10131.70 2:10111.20 3:10131.70 4:10142.00 5:10131.70 6:10101.00 7:9990.01  
 8:9900.99 9:9920.63 10:9900.99  
 9910.80 1:10111.20 2:10131.70 3:10142.00 4:10131.70 5:10101.00 6:9990.01 7:9900.99  
 8:9920.63 9:9900.99 10:9910.80

9910.80 1:10131.70 2:10142.00 3:10131.70 4:10101.00 5:9990.01 6:9900.99 7:9920.63  
8:9900.99 9:9910.80 10:9910.80  
9891.20 1:10142.00 2:10131.70 3:10101.00 4:9990.01 5:9900.99 6:9920.63 7:9900.99  
8:9910.80 9:9910.80 10:9910.80  
9794.32 1:10131.70 2:10101.00 3:9990.01 4:9900.99 5:9920.63 6:9900.99 7:9910.80  
8:9910.80 9:9910.80 10:9891.20  
9615.38 1:10101.00 2:9990.01 3:9900.99 4:9920.63 5:9900.99 6:9910.80 7:9910.80  
8:9910.80 9:9891.20 10:9794.32  
9596.93 1:9990.01 2:9900.99 3:9920.63 4:9900.99 5:9910.80 6:9910.80 7:9910.80  
8:9891.20 9:9794.32 10:9615.38  
9718.17 1:9900.99 2:9920.63 3:9900.99 4:9910.80 5:9910.80 6:9910.80 7:9891.20  
8:9794.32 9:9615.38 10:9596.93  
9718.17 1:9920.63 2:9900.99 3:9910.80 4:9910.80 5:9910.80 6:9891.20 7:9794.32  
8:9615.38 9:9596.93 10:9718.17  
9699.32 1:9900.99 2:9910.80 3:9910.80 4:9910.80 5:9891.20 6:9794.32 7:9615.38  
8:9596.93 9:9718.17 10:9718.17  
9680.54 1:9910.80 2:9910.80 3:9910.80 4:9891.20 5:9794.32 6:9615.38 7:9596.93  
8:9718.17 9:9718.17 10:9699.32  
9689.92 1:9910.80 2:9910.80 3:9891.20 4:9794.32 5:9615.38 6:9596.93 7:9718.17  
8:9718.17 9:9699.32 10:9680.54  
9661.84 1:9910.80 2:9891.20 3:9794.32 4:9615.38 5:9596.93 6:9718.17 7:9718.17  
8:9699.32 9:9680.54 10:9689.92  
9689.92 1:9891.20 2:9794.32 3:9615.38 4:9596.93 5:9718.17 6:9718.17 7:9699.32  
8:9680.54 9:9689.92 10:9661.84  
9689.92 1:9794.32 2:9615.38 3:9596.93 4:9718.17 5:9718.17 6:9699.32 7:9680.54  
8:9689.92 9:9661.84 10:9689.92  
9689.92 1:9615.38 2:9596.93 3:9718.17 4:9718.17 5:9699.32 6:9680.54 7:9689.92  
8:9661.84 9:9689.92 10:9689.92  
9727.63 1:9596.93 2:9718.17 3:9718.17 4:9699.32 5:9680.54 6:9689.92 7:9661.84  
8:9689.92 9:9689.92 10:9689.92  
9746.59 1:9718.17 2:9718.17 3:9699.32 4:9680.54 5:9689.92 6:9661.84 7:9689.92  
8:9689.92 9:9689.92 10:9727.63

### Data set prediksi nilai tukar Rupiah (1 Oktober 2009-31 Oktober 2009)

9661.84 1:9718.17 2:9699.32 3:9680.54 4:9689.92 5:9661.84 6:9689.92 7:9689.92 8:9689.92 9:9727.63  
10:9746.59

9699.32 1:9699.32 2:9680.54 3:9689.92 4:9661.84 5:9689.92 6:9689.92 7:9689.92 8:9727.63 9:9746.59  
10:9661.84

9699.32 1:9680.54 2:9689.92 3:9661.84 4:9689.92 5:9689.92 6:9689.92 7:9727.63 8:9746.59 9:9661.84  
10:9699.32

9643.20 1:9689.92 2:9661.84 3:9689.92 4:9689.92 5:9689.92 6:9727.63 7:9746.59 8:9661.84 9:9699.32  
10:9699.32

9643.20 1:9661.84 2:9689.92 3:9689.92 4:9689.92 5:9727.63 6:9746.59 7:9661.84 8:9699.32 9:9699.32  
10:9643.20

9615.38 1:9689.92 2:9689.92 3:9689.92 4:9727.63 5:9746.59 6:9661.84 7:9699.32 8:9699.32 9:9643.20  
10:9643.20

9551.10 1:9689.92 2:9689.92 3:9727.63 4:9746.59 5:9661.84 6:9699.32 7:9699.32 8:9643.20 9:9643.20  
10:9615.38

9532.89 1:9689.92 2:9727.63 3:9746.59 4:9661.84 5:9699.32 6:9699.32 7:9643.20 8:9643.20 9:9615.38  
10:9551.10

9398.50 1:9727.63 2:9746.59 3:9661.84 4:9699.32 5:9699.32 6:9643.20 7:9643.20 8:9615.38 9:9551.10  
10:9532.89

9425.07 1:9746.59 2:9661.84 3:9699.32 4:9699.32 5:9643.20 6:9643.20 7:9615.38 8:9551.10 9:9532.89  
10:9398.50

9425.07 1:9661.84 2:9699.32 3:9699.32 4:9643.20 5:9643.20 6:9615.38 7:9551.10 8:9532.89 9:9398.50  
10:9425.07

9425.07 1:9699.32 2:9699.32 3:9643.20 4:9643.20 5:9615.38 6:9551.10 7:9532.89 8:9398.50 9:9425.07  
10:9425.07

9398.50 1:9699.32 2:9643.20 3:9643.20 4:9615.38 5:9551.10 6:9532.89 7:9398.50 8:9425.07 9:9425.07  
10:9425.07

9363.30 1:9643.20 2:9643.20 3:9615.38 4:9551.10 5:9532.89 6:9398.50 7:9425.07 8:9425.07 9:9425.07  
10:9398.50

9345.79 1:9643.20 2:9615.38 3:9551.10 4:9532.89 5:9398.50 6:9425.07 7:9425.07 8:9425.07 9:9398.50  
10:9363.30

9363.30 1:9615.38 2:9551.10 3:9532.89 4:9398.50 5:9425.07 6:9425.07 7:9425.07 8:9398.50 9:9363.30  
10:9345.79

9372.07 1:9551.10 2:9532.89 3:9398.50 4:9425.07 5:9425.07 6:9425.07 7:9398.50 8:9363.30 9:9345.79  
10:9363.30

9398.50 1:9532.89 2:9398.50 3:9425.07 4:9425.07 5:9425.07 6:9398.50 7:9363.30 8:9345.79 9:9363.30  
10:9372.07

9398.50 1:9398.50 2:9425.07 3:9425.07 4:9425.07 5:9398.50 6:9363.30 7:9345.79 8:9363.30 9:9372.07  
10:9398.50

9389.67 1:9425.07 2:9425.07 3:9425.07 4:9398.50 5:9363.30 6:9345.79 7:9363.30 8:9372.07 9:9398.50  
10:9398.50

9389.67 1:9425.07 2:9425.07 3:9398.50 4:9363.30 5:9345.79 6:9363.30 7:9372.07 8:9398.50 9:9398.50  
10:9389.67

9354.54 1:9425.07 2:9398.50 3:9363.30 4:9345.79 5:9363.30 6:9372.07 7:9398.50 8:9398.50 9:9389.67  
10:9389.67

9372.07 1:9398.50 2:9363.30 3:9345.79 4:9363.30 5:9372.07 6:9398.50 7:9398.50 8:9389.67 9:9389.67  
10:9354.54

9425.07 1:9363.30 2:9345.79 3:9363.30 4:9372.07 5:9398.50 6:9398.50 7:9389.67 8:9389.67 9:9354.54  
10:9372.07  
9433.96 1:9345.79 2:9363.30 3:9372.07 4:9398.50 5:9398.50 6:9389.67 7:9389.67 8:9354.54 9:9372.07  
10:9425.07  
9433.96 1:9363.30 2:9372.07 3:9398.50 4:9398.50 5:9389.67 6:9389.67 7:9354.54 8:9372.07 9:9425.07  
10:9433.96  
9451.80 1:9372.07 2:9398.50 3:9398.50 4:9389.67 5:9389.67 6:9354.54 7:9372.07 8:9425.07 9:9433.96  
10:9433.96  
9532.89 1:9398.50 2:9398.50 3:9389.67 4:9389.67 5:9354.54 6:9372.07 7:9425.07 8:9433.96 9:9433.96  
10:9451.80  
9596.93 1:9398.50 2:9389.67 3:9389.67 4:9354.54 5:9372.07 6:9425.07 7:9433.96 8:9433.96 9:9451.80  
10:9532.89  
9624.64 1:9389.67 2:9389.67 3:9354.54 4:9372.07 5:9425.07 6:9433.96 7:9433.96 8:9451.80 9:9532.89  
10:9596.93  
9606.15 1:9389.67 2:9354.54 3:9372.07 4:9425.07 5:9433.96 6:9433.96 7:9451.80 8:9532.89 9:9596.93  
10:9624.64

# LAMPIRAN III

## CONTOH FILE MODEL

```

svm_type epsilon_svr
kernel_type rbf
gamma 1.0E-7
nr_class 2
total_sv 178
rho 210.2187319968146
SV
215443.469 1:46.34 2:48.81 3:48.58 4:42.63 5:41.70 6:40.83 7:37.59 8:37.78 9:37.28 10:35.40
215443.469 1:48.81 2:48.58 3:42.63 4:41.70 5:40.83 6:37.59 7:37.78 8:37.28 9:35.40 10:36.51
215443.469 1:48.58 2:42.63 3:41.70 4:40.83 5:37.59 6:37.78 7:37.28 8:35.40 9:36.51 10:38.74
-215443.469 1:42.63 2:41.70 3:40.83 4:37.59 5:37.78 6:37.28 7:35.40 8:36.51 9:38.74 10:42.25
215443.469 1:41.70 2:40.83 3:37.59 4:37.78 5:37.28 6:35.40 7:36.51 8:38.74 9:42.25 10:42.27
215443.469 1:40.83 2:37.59 3:37.78 4:37.28 5:35.40 6:36.51 7:38.74 8:42.25 9:42.27 10:45.47
-215443.469 1:37.59 2:37.78 3:37.28 4:35.40 5:36.51 6:38.74 7:42.25 8:42.27 9:45.47 10:45.73
-168837.91579369034 1:37.78 2:37.28 3:35.40 4:36.51 5:38.74 6:42.25 7:42.27 8:45.47 9:45.73
10:41.58
-215443.469 1:37.28 2:35.40 3:36.51 4:38.74 5:42.25 6:42.27 7:45.47 8:45.73 9:41.58 10:42.16
-182051.37564198885 1:35.40 2:36.51 3:38.74 4:42.25 5:42.27 6:45.47 7:45.73 8:41.58 9:42.16
10:41.44
-215443.469 1:36.51 2:38.74 3:42.25 4:42.27 5:45.47 6:45.73 7:41.58 8:42.16 9:41.44 10:41.68
215443.469 1:38.74 2:42.25 3:42.27 4:45.47 5:45.73 6:41.58 7:42.16 8:41.44 9:41.68 10:40.08
-215443.469 1:42.25 2:42.27 3:45.47 4:45.73 5:41.58 6:42.16 7:41.44 8:41.68 9:40.08 10:40.78
215443.469 1:42.27 2:45.47 3:45.73 4:41.58 5:42.16 6:41.44 7:41.68 8:40.08 9:40.78 10:40.32
-215443.469 1:45.47 2:45.73 3:41.58 4:42.16 5:41.44 6:41.68 7:40.08 8:40.78 9:40.32 10:41.17
-215443.469 1:45.73 2:41.58 3:42.16 4:41.44 5:41.68 6:40.08 7:40.78 8:40.32 9:41.17 10:40.17
-215443.469 1:41.58 2:42.16 3:41.44 4:41.68 5:40.08 6:40.78 7:40.32 8:41.17 9:40.17 10:39.56
-215443.469 1:42.16 2:41.44 3:41.68 4:40.08 5:40.78 6:40.32 7:41.17 8:40.17 9:39.56 10:37.55
-215443.469 1:41.44 2:41.68 3:40.08 4:40.78 5:40.32 6:41.17 7:40.17 8:39.56 9:37.55 10:35.94
215443.469 1:41.68 2:40.08 3:40.78 4:40.32 5:41.17 6:40.17 7:39.56 8:37.55 9:35.94 10:33.98
-215443.469 1:40.08 2:40.78 3:40.32 4:41.17 5:40.17 6:39.56 7:37.55 8:35.94 9:33.98 10:37.51
-215443.469 1:40.78 2:40.32 3:41.17 4:40.17 5:39.56 6:37.55 7:35.94 8:33.98 9:37.51 10:34.93
215443.469 1:40.32 2:41.17 3:40.17 4:39.56 5:37.55 6:35.94 7:33.98 8:37.51 9:34.93 10:34.62
-215443.469 1:41.17 2:40.17 3:39.56 4:37.55 5:35.94 6:33.98 7:37.51 8:34.93 9:34.62 10:39.48
-215443.469 1:40.17 2:39.56 3:37.55 4:35.94 5:33.98 6:37.51 7:34.93 8:34.62 9:39.48 10:38.94
215443.469 1:39.56 2:37.55 3:35.94 4:33.98 5:37.51 6:34.93 7:34.62 8:39.48 9:38.94 10:37.59
215443.469 1:37.55 2:35.94 3:33.98 4:37.51 5:34.93 6:34.62 7:39.48 8:38.94 9:37.59 10:38.96
215443.469 1:35.94 2:33.98 3:37.51 4:34.93 5:34.62 6:39.48 7:38.94 8:37.59 9:38.96 10:42.50
88757.32557046357 1:33.98 2:37.51 3:34.93 4:34.62 5:39.48 6:38.94 7:37.59 8:38.96 9:42.50
10:45.22
-215443.469 1:37.51 2:34.93 3:34.62 4:39.48 5:38.94 6:37.59 7:38.96 8:42.50 9:45.22 10:44.76
215443.469 1:34.93 2:34.62 3:39.48 4:38.94 5:37.59 6:38.96 7:42.50 8:45.22 9:44.76 10:40.15
215443.469 1:34.62 2:39.48 3:38.94 4:37.59 5:38.96 6:42.50 7:45.22 8:44.76 9:40.15 10:41.65
-215443.469 1:39.48 2:38.94 3:37.59 4:38.96 5:42.50 6:45.22 7:44.76 8:40.15 9:41.65 10:45.38
215443.469 1:38.94 2:37.59 3:38.96 4:42.50 5:45.22 6:44.76 7:40.15 8:41.65 9:45.38 10:43.61
215443.469 1:37.59 2:38.96 3:42.50 4:45.22 5:44.76 6:40.15 7:41.65 8:45.38 9:43.61 10:45.52
-215443.469 1:38.96 2:42.50 3:45.22 4:44.76 5:40.15 6:41.65 7:45.38 8:43.61 9:45.52 10:47.07
-215443.469 1:42.50 2:45.22 3:44.76 4:40.15 5:41.65 6:45.38 7:43.61 8:45.52 9:47.07 10:45.71
215443.469 1:45.22 2:44.76 3:40.15 4:41.65 5:45.38 6:43.61 7:45.52 8:47.07 9:45.71 10:42.33
-215443.469 1:44.76 2:40.15 3:41.65 4:45.38 5:43.61 6:45.52 7:47.07 8:45.71 9:42.33 10:47.03
215443.469 1:40.15 2:41.65 3:45.38 4:43.61 5:45.52 6:47.07 7:45.71 8:42.33 9:47.03 10:46.25
215443.469 1:41.65 2:45.38 3:43.61 4:45.52 5:47.07 6:45.71 7:42.33 8:47.03 9:46.25 10:47.35
-215443.469 1:45.38 2:43.61 3:45.52 4:47.07 5:45.71 6:42.33 7:47.03 8:46.25 9:47.35 10:49.16
215443.469 1:43.61 2:45.52 3:47.07 4:45.71 5:42.33 6:47.03 7:46.25 8:47.35 9:49.16 10:48.14
-215443.469 1:45.52 2:47.07 3:45.71 4:42.33 5:47.03 6:46.25 7:47.35 8:49.16 9:48.14 10:51.61
215443.469 1:47.07 2:45.71 3:42.33 4:47.03 5:46.25 6:47.35 7:49.16 8:48.14 9:51.61 10:51.06
45828.523124853324 1:45.71 2:42.33 3:47.03 4:46.25 5:47.35 6:49.16 7:48.14 8:51.61 9:51.06
10:53.0
-215443.469 1:42.33 2:47.03 3:46.25 4:47.35 5:49.16 6:48.14 7:51.61 8:51.06 9:53.00 10:53.48
215443.469 1:47.03 2:46.25 3:47.35 4:49.16 5:48.14 6:51.61 7:51.06 8:53.00 9:53.48 10:52.27
-215443.469 1:46.25 2:47.35 3:49.16 4:48.14 5:51.61 6:51.06 7:53.00 8:53.48 9:52.27 10:54.34
-215443.469 1:47.35 2:49.16 3:48.14 4:51.61 5:51.06 6:53.00 7:53.48 8:52.27 9:54.34 10:52.38
215443.469 1:49.16 2:48.14 3:51.61 4:51.06 5:53.00 6:53.48 7:52.27 8:54.34 9:52.38 10:48.41

```

-215443.469 1:48.14 2:51.61 3:51.06 4:53.00 5:53.48 6:52.27 7:54.34 8:52.38 9:48.41 10:49.66  
 215443.469 1:51.61 2:51.06 3:53.00 4:53.48 5:52.27 6:54.34 7:52.38 8:48.41 9:49.66 10:48.39  
 -215443.469 1:51.06 2:53.00 3:53.48 4:52.27 5:54.34 6:52.38 7:48.41 8:49.66 9:48.39 10:52.64  
 -215443.469 1:53.00 2:53.48 3:52.27 4:54.34 5:52.38 6:48.41 7:49.66 8:48.39 9:52.64 10:52.51  
 -215443.469 1:53.48 2:52.27 3:54.34 4:52.38 5:48.41 6:49.66 7:48.39 8:52.64 9:52.51 10:51.05  
 -215443.469 1:52.27 2:54.34 3:52.38 4:48.41 5:49.66 6:48.39 7:52.64 8:52.51 9:51.05 10:49.15  
 215443.469 1:54.34 2:52.38 3:48.41 4:49.66 5:48.39 6:52.64 7:52.51 8:51.05 9:49.15 10:49.38  
 -215443.469 1:52.38 2:48.41 3:49.66 4:48.39 5:52.64 6:52.51 7:51.05 8:49.15 9:49.38 10:52.24  
 -215443.469 1:48.41 2:49.66 3:48.39 4:52.64 5:52.51 6:51.05 7:49.15 8:49.38 9:52.24 10:50.05  
 -215443.469 1:49.66 2:48.39 3:52.64 4:52.51 5:51.05 6:49.15 7:49.38 8:52.24 9:50.05 10:49.41  
 215443.469 1:48.39 2:52.64 3:52.51 4:51.05 5:49.15 6:49.38 7:52.24 8:50.05 9:49.41 10:49.25  
 -215443.469 1:52.64 2:52.51 3:51.05 4:49.15 5:49.38 6:52.24 7:50.05 8:49.41 9:49.25 10:49.98  
 -215443.469 1:52.51 2:51.05 3:49.15 4:49.38 5:52.24 6:50.05 7:49.41 8:49.25 9:49.98 10:50.33  
 -215443.469 1:51.05 2:49.15 3:49.38 4:52.24 5:50.05 6:49.41 7:49.25 8:49.98 9:50.33 10:45.88  
 215443.469 1:49.15 2:49.38 3:52.24 4:50.05 5:49.41 6:49.25 7:49.98 8:50.33 9:45.88 10:46.51  
 215443.469 1:49.38 2:52.24 3:50.05 4:49.41 5:49.25 6:49.98 7:50.33 8:45.88 9:46.51 10:47.35  
 215443.469 1:52.24 2:50.05 3:49.41 4:49.25 5:49.98 6:50.33 7:45.88 8:46.51 9:47.35 10:48.62  
 -215443.469 1:50.05 2:49.41 3:49.25 4:49.98 5:50.33 6:45.88 7:46.51 8:47.35 9:48.62 10:50.80  
 -215443.469 1:49.41 2:49.25 3:49.98 4:50.33 5:45.88 6:46.51 7:47.35 8:48.62 9:50.80 10:50.14  
 215443.469 1:49.25 2:49.98 3:50.33 4:55.88 5:46.51 6:47.35 7:48.62 8:50.80 9:50.14 10:49.92  
 -177830.3415224753 1:49.98 2:50.33 3:45.88 4:46.51 5:47.35 6:48.62 7:50.80 8:50.14 9:49.92  
 10:50.97  
 215443.469 1:50.33 2:45.88 3:46.51 4:47.35 5:48.62 6:50.80 7:50.14 8:49.92 9:50.97 10:51.12  
 215443.469 1:45.88 2:46.51 3:47.35 4:48.62 5:50.80 6:50.14 7:49.92 8:50.97 9:51.12 10:53.20  
 -215443.469 1:46.51 2:47.35 3:48.62 4:50.80 5:50.14 6:49.92 7:50.97 8:51.12 9:53.20 10:54.47  
 215443.469 1:47.35 2:48.62 3:50.80 4:50.14 5:49.92 6:50.97 7:51.12 8:53.20 9:54.47 10:53.84  
 215443.469 1:48.62 2:50.80 3:50.14 4:49.92 5:50.97 6:51.12 7:53.20 8:54.47 9:53.84 10:56.34  
 215443.469 1:50.80 2:50.14 3:49.92 4:50.97 5:51.12 6:53.20 7:54.47 8:53.84 9:56.34 10:56.71  
 -215443.469 1:50.14 2:49.92 3:50.97 4:51.12 5:53.20 6:54.47 7:53.84 8:56.34 9:56.71 10:58.63  
 215443.469 1:49.92 2:50.97 3:51.12 4:53.20 5:54.47 6:53.84 7:56.34 8:56.71 9:58.63 10:58.5  
 -215443.469 1:50.97 2:51.12 3:53.20 4:54.47 5:53.84 6:56.34 7:56.71 8:58.63 9:58.5 10:58.85  
 215443.469 1:51.12 2:53.20 3:54.47 4:53.84 5:56.34 6:56.71 7:58.63 8:58.5 9:58.85 10:58.02  
 -215443.469 1:53.20 2:54.47 3:53.84 4:56.34 5:56.71 6:58.63 7:58.5 8:58.85 9:58.02 10:58.62  
 215443.469 1:54.47 2:53.84 3:56.34 4:56.71 5:58.63 6:58.5 7:58.85 8:58.02 9:58.62 10:56.34  
 215443.469 1:53.84 2:56.34 3:56.71 4:58.63 5:58.5 6:58.85 7:58.02 8:58.62 9:56.34 10:59.03  
 215443.469 1:56.34 2:56.71 3:58.63 4:58.5 5:58.85 6:58.02 7:58.62 8:56.34 9:59.03 10:59.65  
 -215443.469 1:56.71 2:58.63 3:58.5 4:58.85 5:58.02 6:58.62 7:56.34 8:59.03 9:59.65 10:62.02  
 -100304.80070945075 1:58.63 2:58.5 3:58.85 4:58.02 5:58.62 6:56.34 7:59.03 8:59.65 9:62.02  
 10:60.55  
 215443.469 1:58.5 2:58.85 3:58.02 4:58.62 5:56.34 6:59.03 7:59.65 8:62.02 9:60.55 10:61.17  
 215443.469 1:58.85 2:58.02 3:58.62 4:56.34 5:59.03 6:59.65 7:62.02 8:60.55 9:61.17 10:62.45  
 215443.469 1:58.02 2:58.62 3:56.34 4:59.03 5:59.65 6:62.02 7:60.55 8:61.17 9:62.45 10:63.45  
 215443.469 1:58.62 2:56.34 3:59.03 4:59.65 5:62.02 6:60.55 7:61.17 8:62.45 9:63.45 10:65.08  
 215443.469 1:56.34 2:59.03 3:59.65 4:62.02 5:60.55 6:61.17 7:62.45 8:63.45 9:65.08 10:66.31  
 148695.75129812385 1:59.03 2:59.65 3:62.02 4:60.55 5:61.17 6:62.45 7:63.45 8:65.08 9:66.31  
 10:68.58  
 -215443.469 1:59.65 2:62.02 3:60.55 4:61.17 5:62.45 6:63.45 7:65.08 8:66.31 9:68.58 10:68.55  
 215443.469 1:62.02 2:60.55 3:61.17 4:62.45 5:63.45 6:65.08 7:66.31 8:68.58 9:68.55 10:66.12  
 -215443.469 1:60.55 2:61.17 3:62.45 4:63.45 5:65.08 6:66.31 7:68.58 8:68.55 9:66.12 10:68.81  
 -215443.469 1:61.17 2:62.45 3:63.45 4:65.08 5:66.31 6:68.58 7:68.55 8:66.12 9:68.81 10:68.44  
 215443.469 1:62.45 2:63.45 3:65.08 4:66.31 5:68.58 6:68.55 7:66.12 8:68.81 9:68.44 10:68.09  
 215443.469 1:63.45 2:65.08 3:66.31 4:68.58 5:68.55 6:66.12 7:68.81 8:68.44 9:68.09 10:70.01  
 215443.469 1:65.08 2:66.31 3:68.58 4:68.55 5:66.12 6:68.81 7:68.44 8:68.09 9:70.01 10:71.33  
 -215443.469 1:66.31 2:68.58 3:68.55 4:66.12 5:68.81 6:68.44 7:68.09 8:70.01 9:71.33 10:72.68  
 -215443.469 1:68.58 2:68.55 3:66.12 4:68.81 5:68.44 6:68.09 7:70.01 8:71.33 9:72.68 10:72.04  
 -215443.469 1:68.55 2:66.12 3:68.81 4:68.44 5:68.09 6:70.01 7:71.33 8:72.68 9:72.04 10:70.62  
 215443.469 1:66.12 2:68.81 3:68.44 4:68.09 5:70.01 6:71.33 7:72.68 8:72.04 9:70.62 10:70.47  
 215443.469 1:68.81 2:68.44 3:68.09 4:70.01 5:71.33 6:72.68 7:72.04 8:70.62 9:70.47 10:71.03  
 -215443.469 1:68.44 2:68.09 3:70.01 4:71.33 5:72.68 6:72.04 7:70.62 8:70.47 9:71.03 10:71.37  
 -215443.469 1:68.09 2:70.01 3:71.33 4:72.68 5:72.04 6:70.62 7:70.47 8:71.03 9:71.37 10:69.55  
 215443.469 1:70.01 2:71.33 3:72.68 4:72.04 5:70.62 6:70.47 7:71.03 8:71.37 9:69.55 10:66.93  
 -5628.648001408503 1:71.33 2:72.68 3:72.04 4:70.62 5:70.47 6:71.03 7:71.37 8:69.55 9:66.93  
 10:68.79  
 215443.469 1:72.68 2:72.04 3:70.62 4:70.47 5:71.03 6:71.37 7:69.55 8:66.93 9:68.79 10:68.67  
 -215443.469 1:72.04 2:70.62 3:70.47 4:71.03 5:71.37 6:69.55 7:66.93 8:68.79 9:68.67 10:70.23  
 215443.469 1:70.62 2:70.47 3:71.03 4:71.37 5:69.55 6:66.93 7:68.79 8:68.67 9:70.23 10:69.16  
 -215443.469 1:70.47 2:71.03 3:71.37 4:69.55 5:66.93 6:68.79 7:68.67 8:70.23 9:69.16 10:71.49  
 -215443.469 1:71.03 2:71.37 3:69.55 4:66.93 5:68.79 6:68.67 7:70.23 8:69.16 9:71.49 10:69.89  
 -215443.469 1:71.37 2:69.55 3:66.93 4:68.79 5:68.67 6:70.23 7:69.16 8:71.49 9:69.89 10:69.31  
 -215443.469 1:69.55 2:66.93 3:68.79 4:68.67 5:70.23 6:69.16 7:71.49 8:69.89 9:69.31 10:66.73  
 -215443.469 1:66.93 2:68.79 3:68.67 4:70.23 5:69.16 6:71.49 7:69.89 8:69.31 9:66.73 10:64.05

-215443.469 1:68.79 2:68.67 3:70.23 4:69.16 5:71.49 6:69.89 7:69.31 8:66.73 9:64.05 10:62.93  
 -82559.18285461473 1:68.67 2:70.23 3:69.16 4:71.49 5:69.89 6:69.31 7:66.73 8:64.05 9:62.93  
 10:60.14  
 -215443.469 1:70.23 2:69.16 3:71.49 4:69.89 5:69.31 6:66.73 7:64.05 8:62.93 9:60.14 10:60.41  
 -215443.469 1:69.16 2:71.49 3:69.89 4:69.31 5:66.73 6:64.05 7:62.93 8:60.14 9:60.41 10:59.89  
 -215443.469 1:71.49 2:69.89 3:69.31 4:66.73 5:64.05 6:62.93 7:60.14 8:60.41 9:59.89 10:59.69  
 215443.469 1:69.89 2:69.31 3:66.73 4:64.05 5:62.93 6:60.14 7:60.41 8:59.89 9:59.69 10:59.52  
 215443.469 1:69.31 2:66.73 3:64.05 4:62.93 5:60.14 6:60.41 7:59.89 8:59.69 9:59.52 10:61.54  
 215443.469 1:66.73 2:64.05 3:62.93 4:60.14 5:60.41 6:59.89 7:59.69 8:59.52 9:61.54 10:62.02  
 215443.469 1:64.05 2:62.93 3:60.14 4:60.41 5:59.89 6:59.69 7:59.52 8:61.54 9:62.02 10:63.56  
 215443.469 1:62.93 2:60.14 3:60.41 4:59.89 5:59.69 6:59.52 7:61.54 8:62.02 9:63.56 10:63.98  
 -215443.469 1:60.14 2:60.41 3:59.89 4:59.69 5:59.52 6:61.54 7:62.02 8:63.56 9:63.98 10:64.72  
 215443.469 1:60.41 2:59.89 3:59.69 4:59.52 5:61.54 6:62.02 7:63.56 8:63.98 9:64.72 10:63.57  
 215443.469 1:59.89 2:59.69 3:59.52 4:61.54 5:62.02 6:63.56 7:63.98 8:64.72 9:63.57 10:66.11  
 215443.469 1:59.69 2:59.52 3:61.54 4:62.02 5:63.56 6:63.98 7:64.72 8:63.57 9:66.11 10:66.55  
 -215443.469 1:59.52 2:61.54 3:62.02 4:63.56 5:63.98 6:64.72 7:63.57 8:66.11 9:66.55 10:68.38  
 -215443.469 1:61.54 2:62.02 3:63.56 4:63.98 5:64.72 6:63.57 7:66.11 8:66.55 9:68.38 10:67.23  
 215443.469 1:62.02 2:63.56 3:63.98 4:64.72 5:63.57 6:66.11 7:66.55 8:68.38 9:67.23 10:63.35  
 215443.469 1:63.56 2:63.98 3:64.72 4:63.57 5:66.11 6:66.55 7:68.38 8:67.23 9:63.35 10:66.94  
 215443.469 1:63.98 2:64.72 3:63.57 4:66.11 5:66.55 6:68.38 7:67.23 8:63.35 9:66.94 10:69.45  
 -37485.738224350665 1:64.72 2:63.57 3:66.11 4:66.55 5:68.38 6:67.23 7:63.35 8:66.94 9:69.45  
 10:71.58  
 215443.469 1:63.57 2:66.11 3:66.55 4:68.38 5:67.23 6:63.35 7:66.94 8:69.45 9:71.58 10:71.42  
 -215443.469 1:66.11 2:66.55 3:68.38 4:67.23 5:63.35 6:66.94 7:69.45 8:71.58 9:71.42 10:71.97  
 -215443.469 1:66.55 2:68.38 3:67.23 4:63.35 5:66.94 6:69.45 7:71.58 8:71.42 9:71.97 10:71.94  
 -215443.469 1:68.38 2:67.23 3:63.35 4:66.94 5:69.45 6:71.58 7:71.42 8:71.97 9:71.94 10:70.93  
 -215443.469 1:67.23 2:63.35 3:66.94 4:69.45 5:71.58 6:71.42 7:71.97 8:71.94 9:70.93 10:70.60  
 215443.469 1:63.35 2:66.94 3:69.45 4:71.58 5:71.42 6:71.97 7:71.94 8:70.93 9:70.60 10:69.45  
 215443.469 1:66.94 2:69.45 3:71.58 4:71.42 5:71.97 6:71.94 7:70.93 8:70.60 9:69.45 10:70.16  
 -215443.469 1:69.45 2:71.58 3:71.42 4:71.97 5:71.94 6:70.93 7:70.60 8:69.45 9:70.16 10:70.52  
 -215443.469 1:71.58 2:71.42 3:71.97 4:71.94 5:70.93 6:70.60 7:69.45 8:70.16 9:70.52 10:67.51  
 215443.469 1:71.42 2:71.97 3:71.94 4:70.93 5:70.60 6:69.45 7:70.16 8:70.52 9:67.51 10:66.75  
 215443.469 1:71.97 2:71.94 3:70.93 4:70.60 5:69.45 6:70.16 7:70.52 8:67.51 9:66.75 10:69.19  
 215443.469 1:71.94 2:70.93 3:70.60 4:69.45 5:70.16 6:70.52 7:67.51 8:66.75 9:69.19 10:72.42  
 215443.469 1:70.93 2:70.60 3:69.45 4:70.16 5:70.52 6:67.51 7:66.75 8:69.19 9:72.42 10:72.54  
 215443.469 1:70.60 2:69.45 3:70.16 4:70.52 5:71.42 6:66.75 7:69.19 8:72.42 9:72.54 10:73.19  
 -215443.469 1:69.45 2:70.16 3:70.52 4:67.51 5:66.75 6:69.19 7:72.42 8:72.54 9:73.19 10:73.82  
 -215443.469 1:70.16 2:70.52 3:67.51 4:66.75 5:69.19 6:72.42 7:72.54 8:73.19 9:73.82 10:71.55  
 215443.469 1:70.52 2:67.51 3:66.75 4:69.19 5:72.42 6:72.54 7:73.19 8:73.82 9:71.55 10:71.43  
 215443.469 1:67.51 2:66.75 3:69.19 4:72.42 5:72.54 6:73.19 7:73.82 8:71.55 9:71.43 10:72.49  
 -215443.469 1:66.75 2:69.19 3:72.42 4:72.54 5:73.19 6:73.82 7:71.55 8:71.43 9:72.49 10:72.79  
 -215443.469 1:69.19 2:72.42 3:72.54 4:73.19 5:73.82 6:71.55 7:71.43 8:72.49 9:72.79 10:69.96  
 -215443.469 1:72.42 2:72.54 3:73.19 4:73.82 5:71.55 6:71.43 7:72.49 8:72.79 9:69.96 10:68.05  
 -215443.469 1:72.54 2:73.19 3:73.82 4:71.55 5:71.43 6:72.49 7:72.79 8:69.96 9:68.05 10:68.05  
 215443.469 1:73.19 2:73.82 3:71.55 4:71.43 5:72.49 6:72.79 7:69.96 8:68.05 9:68.05 10:67.96  
 215443.469 1:73.82 2:71.55 3:71.43 4:72.49 5:72.79 6:69.96 7:68.05 8:68.05 9:67.96 10:68.02  
 215443.469 1:71.55 2:71.43 3:72.49 4:72.79 5:69.96 6:68.05 7:68.05 8:67.96 9:68.02 10:71.10  
 215443.469 1:71.43 2:72.49 3:72.79 4:69.96 5:68.05 6:68.05 7:67.96 8:68.02 9:71.10 10:71.31  
 -215443.469 1:72.49 2:72.79 3:69.96 4:68.05 5:68.05 6:67.96 7:68.02 8:71.10 9:71.31 10:71.94  
 -215443.469 1:72.79 2:69.96 3:68.05 4:68.05 5:67.96 6:68.02 7:71.10 8:71.31 9:71.94 10:69.29  
 215443.469 1:69.96 2:68.05 3:68.05 4:67.96 5:68.02 6:71.10 7:71.31 8:71.94 9:69.29 10:68.86  
 215443.469 1:68.05 2:68.05 3:67.96 4:68.02 5:71.10 6:71.31 7:71.94 8:69.29 9:68.86 10:70.93  
 215443.469 1:68.05 2:67.96 3:68.02 4:71.10 5:71.31 6:71.94 7:69.29 8:68.86 9:70.93 10:72.51  
 -215443.469 1:67.96 2:68.02 3:71.10 4:71.31 5:71.94 6:69.29 7:68.86 8:70.93 9:72.51 10:72.47  
 -215443.469 1:68.02 2:71.10 3:71.31 4:71.94 5:69.29 6:68.86 7:70.93 8:72.51 9:72.47 10:72.04  
 215443.469 1:71.10 2:71.31 3:71.94 4:69.29 5:68.86 6:70.93 7:72.51 8:72.47 9:72.04 10:69.71  
 -215443.469 1:71.31 2:71.94 3:69.29 4:68.86 5:70.93 6:72.51 7:72.47 8:72.04 9:69.71 10:71.55  
 -215443.469 1:71.94 2:69.29 3:68.86 4:70.93 5:72.51 6:72.47 7:72.04 8:69.71 9:71.55 10:68.97  
 -215443.469 1:69.29 2:68.86 3:70.93 4:72.51 5:72.47 6:72.04 7:69.71 8:71.55 9:68.97 10:65.79  
 215443.469 1:68.86 2:70.93 3:72.51 4:72.47 5:72.04 6:69.71 7:71.55 8:68.97 9:65.79 10:66.02  
 -174914.0042454616 1:70.93 2:72.51 3:72.47 4:72.04 5:69.71 6:71.55 7:68.97 8:65.79 9:66.02  
 10:66.84  
 215443.469 1:72.51 2:72.47 3:72.04 4:69.71 5:71.55 6:68.97 7:65.79 8:66.02 9:66.84 10:66.71

## LAMPIRAN IV

### SOURCE CODE SVM\_TRAIN

```

public static svm_model svm_train(svm_problem prob, svm_parameter param)
{
    svm_model model = new svm_model();
    model.param = param;

    if(param.svm_type == svm_parameter.ONE_CLASS ||
        param.svm_type == svm_parameter.EPSILON_SVR ||
        param.svm_type == svm_parameter.NU_SVR)
    {
        // regression or one-class-svm
        model.nr_class = 2;
        model.label = null;
        model.nSV = null;
        model.probA = null; model.probB = null;
        model.sv_coef = new double[1][];

        if(param.probability == 1 &&
            (param.svm_type == svm_parameter.EPSILON_SVR ||
             param.svm_type == svm_parameter.NU_SVR))
        {
            model.probA = new double[1];
            model.probA[0] = svm_svr_probability(prob,param);
        }

        decision_function f = svm_train_one(prob,param,0,0);
        model.rho = new double[1];
        model.rho[0] = f.rho;

        int nSV = 0;
        int i;
        for(i=0;i<prob.l;i++)
            if(Math.abs(f.alpha[i]) > 0) ++nSV;
        model.l = nSV;
        model.SV = new svm_node[nSV][];
        model.sv_coef[0] = new double[nSV];
        int j = 0;
        for(i=0;i<prob.l;i++)
            if(Math.abs(f.alpha[i]) > 0)
            {
                model.SV[j] = prob.x[i];
                model.sv_coef[0][j] = f.alpha[i];
                ++j;
            }
    }
    else
    {
        // classification
        int l = prob.l;
        int[] tmp_nr_class = new int[1];
        int[][] tmp_label = new int[1][];
        int[][] tmp_start = new int[1][];
        int[][] tmp_count = new int[1][];
        int[] perm = new int[1];

        // group training data of the same class

        svm_group_classes(prob,tmp_nr_class,tmp_label,tmp_start,tmp_count,perm);
    }
}

```



```

int nr_class = tmp_nr_class[0];
int[] label = tmp_label[0];
int[] start = tmp_start[0];
int[] count = tmp_count[0];
svm_node[][] x = new svm_node[1][];
int i;
for(i=0;i<1;i++)
    x[i] = prob.x[perm[i]];

// calculate weighted C

double[] weighted_C = new double[nr_class];
for(i=0;i<nr_class;i++)
    weighted_C[i] = param.C;
for(i=0;i<param.nr_weight;i++)
{
    int j;
    for(j=0;j<nr_class;j++)
        if(param.weight_label[i] == label[j])
            break;
    if(j == nr_class)
        System.err.print("warning: class label
"+param.weight_label[i]+" specified in weight is not found\n");
    else
        weighted_C[j] *= param.weight[i];
}

// train k*(k-1)/2 models

boolean[] nonzero = new boolean[1];
for(i=0;i<1;i++)
    nonzero[i] = false;
decision_function[] f = new decision_function[nr_class*(nr_class-
1)/2];

double[] probA=null,probB=null;
if (param.probability == 1)
{
    probA=new double[nr_class*(nr_class-1)/2];
    probB=new double[nr_class*(nr_class-1)/2];
}

int p = 0;
for(i=0;i<nr_class;i++)
    for(int j=i+1;j<nr_class;j++)
    {
        svm_problem sub_prob = new svm_problem();
        int si = start[i], sj = start[j];
        int ci = count[i], cj = count[j];
        sub_prob.l = ci+cj;
        sub_prob.x = new svm_node[sub_prob.l][];
        sub_prob.y = new double[sub_prob.l];
        int k;
        for(k=0;k<ci;k++)
        {
            sub_prob.x[k] = x[si+k];
            sub_prob.y[k] = +1;
        }
        for(k=0;k<cj;k++)
        {
            sub_prob.x[ci+k] = x[sj+k];
            sub_prob.y[ci+k] = -1;
        }
    }

```

```

        if(param.probability == 1)
        {
            double[] probAB=new double[2];

            svm_binary_svc_probability(sub_prob,param,weighted_C[i],weighted_C[j],probAB);
            probA[p]=probAB[0];
            probB[p]=probAB[1];
        }

        f[p] =
svm_train_one(sub_prob,param,weighted_C[i],weighted_C[j]);
        for(k=0;k<ci;k++)
            if(!nonzero[si+k] && Math.abs(f[p].alpha[k]) >
0)
                nonzero[si+k] = true;
        for(k=0;k<cj;k++)
            if(!nonzero[sj+k] &&
Math.abs(f[p].alpha[ci+k]) > 0)
                nonzero[sj+k] = true;
        ++p;
    }

    // build output
    model.nr_class = nr_class;

    model.label = new int[nr_class];
    for(i=0;i<nr_class;i++)
        model.label[i] = label[i];

    model.rho = new double[nr_class*(nr_class-1)/2];
    for(i=0;i<nr_class*(nr_class-1)/2;i++)
        model.rho[i] = f[i].rho;

    if(param.probability == 1)
    {
        model.probA = new double[nr_class*(nr_class-1)/2];
        model.probB = new double[nr_class*(nr_class-1)/2];
        for(i=0;i<nr_class*(nr_class-1)/2;i++)
        {
            model.probA[i] = probA[i];
            model.probB[i] = probB[i];
        }
    }
    else
    {
        model.probA=null;
        model.probB=null;
    }

    int nnz = 0;
    int[] nz_count = new int[nr_class];
    model.nSV = new int[nr_class];
    for(i=0;i<nr_class;i++)
    {
        int nSV = 0;
        for(int j=0;j<count[i];j++)
            if(nonzero[start[i]+j])
            {
                ++nSV;
                ++nnz;
            }
    }

```

```

        model.nSV[i] = nSV;
        nz_count[i] = nSV;
    }

    System.out.print("Total nSV = "+nnz+"\n");

    model.l = nnz;
    model.SV = new svm_node[nnz][];
    p = 0;
    for(i=0;i<l;i++)
        if(nonzero[i]) model.SV[p++] = x[i];

    int[] nz_start = new int[nr_class];
    nz_start[0] = 0;
    for(i=1;i<nr_class;i++)
        nz_start[i] = nz_start[i-1]+nz_count[i-1];

    model.sv_coef = new double[nr_class-1][];
    for(i=0;i<nr_class-1;i++)
        model.sv_coef[i] = new double[nnz];

    p = 0;
    for(i=0;i<nr_class;i++)
        for(int j=i+1;j<nr_class;j++)
        {
            // classifier (i,j): coefficients with
            // i are in sv_coef[j-1][nz_start[i]...],
            // j are in sv_coef[i][nz_start[j]...]

            int si = start[i];
            int sj = start[j];
            int ci = count[i];
            int cj = count[j];

            int q = nz_start[i];
            int k;
            for(k=0;k<ci;k++)
                if(nonzero[si+k])
                    model.sv_coef[j-1][q++] =
f[p].alpha[k];

            q = nz_start[j];
            for(k=0;k<cj;k++)
                if(nonzero[sj+k])
                    model.sv_coef[i][q++] =
f[p].alpha[ci+k];

            ++p;
        }
    }
    return model;
}

```

## LAMPIRAN V

## SOURCE CODE SVM\_PREDICT

```

public static void svm_predict_values(svm_model model, svm_node[] x, double[]
dec_values)
{
    if(model.param.svm_type == svm_parameter.ONE_CLASS ||
        model.param.svm_type == svm_parameter.EPSILON_SVR ||
        model.param.svm_type == svm_parameter.NU_SVR)
    {
        double[] sv_coef = model.sv_coef[0];
        double sum = 0;
        for(int i=0;i<model.l;i++)
            sum += sv_coef[i] *
Kernel.k_function(x,model.SV[i],model.param);
        sum -= model.rho[0];
        dec_values[0] = sum;
    }
    else
    {
        int i;
        int nr_class = model.nr_class;
        int l = model.l;

        double[] kvalue = new double[l];
        for(i=0;i<l;i++)
            kvalue[i] = Kernel.k_function(x,model.SV[i],model.param);

        int[] start = new int[nr_class];
        start[0] = 0;
        for(i=1;i<nr_class;i++)
            start[i] = start[i-1]+model.nSV[i-1];

        int p=0;
        for(i=0;i<nr_class;i++)
            for(int j=i+1;j<nr_class;j++)
            {
                double sum = 0;
                int si = start[i];
                int sj = start[j];
                int ci = model.nSV[i];
                int cj = model.nSV[j];

                int k;
                double[] coef1 = model.sv_coef[j-1];
                double[] coef2 = model.sv_coef[i];
                for(k=0;k<ci;k++)
                    sum += coef1[si+k] * kvalue[si+k];
                for(k=0;k<cj;k++)
                    sum += coef2[sj+k] * kvalue[sj+k];
                sum -= model.rho[p];
                dec_values[p] = sum;
                p++;
            }
    }
}

public static double svm_predict(svm_model model, svm_node[] x)
{
    if(model.param.svm_type == svm_parameter.ONE_CLASS ||

```

```

        model.param.svm_type == svm_parameter.EPSILON_SVR ||
        model.param.svm_type == svm_parameter.NU_SVR)
    {
        double[] res = new double[1];
        svm_predict_values(model, x, res);

        if(model.param.svm_type == svm_parameter.ONE_CLASS)
            return (res[0]>0)?1:-1;
        else
            return res[0];
    }
    else
    {
        int i;
        int nr_class = model.nr_class;
        double[] dec_values = new double[nr_class*(nr_class-1)/2];
        svm_predict_values(model, x, dec_values);

        int[] vote = new int[nr_class];
        for(i=0;i<nr_class;i++)
            vote[i] = 0;
        int pos=0;
        for(i=0;i<nr_class;i++)
            for(int j=i+1;j<nr_class;j++)
            {
                if(dec_values[pos++] > 0)
                    ++vote[i];
                else
                    ++vote[j];
            }

        int vote_max_idx = 0;
        for(i=1;i<nr_class;i++)
            if(vote[i] > vote[vote_max_idx])
                vote_max_idx = i;
        return model.label[vote_max_idx];
    }
}

public static double svm_predict_probability(svm_model model, svm_node[] x,
double[] prob_estimates)
{
    if ((model.param.svm_type == svm_parameter.C_SVC || model.param.svm_type
== svm_parameter.NU_SVC) &&
        model.probA!=null && model.probB!=null)
    {
        int i;
        int nr_class = model.nr_class;
        double[] dec_values = new double[nr_class*(nr_class-1)/2];
        svm_predict_values(model, x, dec_values);

        double min_prob=1e-7;
        double[][] pairwise_prob=new double[nr_class][nr_class];

        int k=0;
        for(i=0;i<nr_class;i++)
            for(int j=i+1;j<nr_class;j++)
            {
                pairwise_prob[i][j]=Math.min(Math.max(sigmoid_predict(dec_values[k],model.probA
[k],model.probB[k]),min_prob),1-min_prob);
                pairwise_prob[j][i]=1-pairwise_prob[i][j];
                k++;
            }
    }
}

```

```
    }  
    multiclass_probability(nr_class, pairwise_prob, prob_estimates);  
  
    int prob_max_idx = 0;  
    for(i=1; i<nr_class; i++)  
        if(prob_estimates[i] > prob_estimates[prob_max_idx])  
            prob_max_idx = i;  
    return model.label[prob_max_idx];  
}  
else  
    return svm_predict(model, x);  
}
```

## LAMPIRAN VI

## SOURCE CODE SVM\_CROSS\_VALIDATION

```

public static void svm_cross_validation(svm_problem prob, svm_parameter param, int
nr_fold, double[] target)
{
    int i;
    int[] fold_start = new int[nr_fold+1];
    int l = prob.l;
    int[] perm = new int[l];

    // stratified cv may not give leave-one-out rate
    // Each class to l folds -> some folds may have zero elements
    if((param.svm_type == svm_parameter.C_SVC ||
        param.svm_type == svm_parameter.NU_SVC) && nr_fold < l)
    {
        int[] tmp_nr_class = new int[1];
        int[][] tmp_label = new int[1][];
        int[][] tmp_start = new int[1][];
        int[][] tmp_count = new int[1][];

        svm_group_classes(prob,tmp_nr_class,tmp_label,tmp_start,tmp_count,perm);

        int nr_class = tmp_nr_class[0];
        int[] label = tmp_label[0];
        int[] start = tmp_start[0];
        int[] count = tmp_count[0];

        // random shuffle and then data grouped by fold using the array
perm
        int[] fold_count = new int[nr_fold];
        int c;
        int[] index = new int[l];
        for(i=0;i<l;i++)
            index[i]=perm[i];
        for (c=0; c<nr_class; c++)
            for(i=0;i<count[c];i++)
            {
                int j = i+(int)(Math.random()*(count[c]-i));
                do {int _=index[start[c]+j];
index[start[c]+j]=index[start[c]+i]; index[start[c]+i]=_;} while(false);
            }
        for(i=0;i<nr_fold;i++)
        {
            fold_count[i] = 0;
            for (c=0; c<nr_class;c++)
                fold_count[i]+=(i+1)*count[c]/nr_fold-
i*count[c]/nr_fold;
        }
        fold_start[0]=0;
        for (i=1;i<=nr_fold;i++)
            fold_start[i] = fold_start[i-1]+fold_count[i-1];
        for (c=0; c<nr_class;c++)
            for(i=0;i<nr_fold;i++)
            {
                int begin = start[c]+i*count[c]/nr_fold;
                int end = start[c]+(i+1)*count[c]/nr_fold;
                for(int j=begin;j<end;j++)
                {

```

```

        perm[fold_start[i]] = index[j];
        fold_start[i]++;
    }
}
fold_start[0]=0;
for (i=1;i<=nr_fold;i++)
    fold_start[i] = fold_start[i-1]+fold_count[i-1];
}
else
{
    for(i=0;i<l;i++) perm[i]=i;
    for(i=0;i<l;i++)
    {
        int j = i+(int)(Math.random()*(l-i));
        do {int _=perm[i]; perm[i]=perm[j]; perm[j]=_;}
        while(false);
    }
    for(i=0;i<=nr_fold;i++)
        fold_start[i]=i*l/nr_fold;
}

for(i=0;i<nr_fold;i++)
{
    int begin = fold_start[i];
    int end = fold_start[i+1];
    int j,k;
    svm_problem subprob = new svm_problem();

    subprob.l = l-(end-begin);
    subprob.x = new svm_node[subprob.l][];
    subprob.y = new double[subprob.l];

    k=0;
    for(j=0;j<begin;j++)
    {
        subprob.x[k] = prob.x[perm[j]];
        subprob.y[k] = prob.y[perm[j]];
        ++k;
    }
    for(j=end;j<l;j++)
    {
        subprob.x[k] = prob.x[perm[j]];
        subprob.y[k] = prob.y[perm[j]];
        ++k;
    }
    svm_model submodel = svm_train(subprob,param);
    if(param.probability==1 &&
        (param.svm_type == svm_parameter.C_SVC ||
         param.svm_type == svm_parameter.NU_SVC))
    {
        double[] prob_estimates= new
double[svm_get_nr_class(submodel)];
        for(j=begin;j<end;j++)
            target[perm[j]] =
svm_predict_probability(submodel,prob.x[perm[j]],prob_estimates);
    }
    else
        for(j=begin;j<end;j++)
            target[perm[j]] =
svm_predict(submodel,prob.x[perm[j]]);
}
}
}

```