Prediksi Pergerakan Indeks Harga Saham Menggunakan *Artificial Neural Network dan Support Vector Machine*

Prediction of Stock Price Index Movement Using Artificial Neural Network and Support Vector Machine

Fani Nuraini[1], Drs. Jondri, M.Si[2], Tjokorda Agung Budi W, ST., MT[3]
[1,2,3]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi no.1, Dayeuh Kolot, Bandung 40257 Indonesia
[1]faninuraini.id@gmail.com, [2]jondri@telkomuniversity.ac.id , [3]cokagung2001@gmail.com

Abstrak

Prediksi pergerakan indeks harga saham dapat dikategorikan sebagai permasalahan yang cukup menantang dalam prediksi finansial. Namun, kompleksitas pasar saham mengakibatkan pengembangan model prediksi yang dapat dikatakan efektif menjadi sangat sulit. Dua model yang digunakan untuk melakukan prediksi, yaitu *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM) dikenal cukup akurat dalam memprediksi pergerakan indeks harga saham. Penelitian menggunakan sepuluh indikator analisis teknikal saham sebagai masukan untuk kedua model yang dibangun. Data indeks saham yang digunakan berasal dari Indeks Harga Saham Gabungan (IHSG). Penelitian menunjukkan bahwa SVM mengungguli ANN dari segi akurasi prediksi pergerakan harga saham IHSG dengan akurasi tertinggi SVM sebesar 56,405% dan akurasi tertinggi ANN sebesar 56,40498% untuk data masukan berupa analisis teknikal saham dengan periode waktu tiga hari.

Kata Kunci: *Prediksi, Artificial Neural Network, Support Vector Machine, Indeks Harga Saham Gabungan, Analisis Teknikal*

Abstract

Stock market movement prediction is one of challenging case on financial prediction problem. A precise prediction can make a big advantage for the investors. But, stock market with its complexity makes a hard searching for development of an effective model Two prediction models: Artificial Neural Network (ANN) dan Support Vector Machine (SVM) known to be accurate for doing a prediction of stock price index movement. This research uses 10 technical analysis indicator of stock market for input of each models using the data from Indeks Harga Saham Gabungan (IHSG). As the result, SVM have a higher accurracy (56,405%) than ANN model (56,40498%) for input based on technical analysis indicator for 3-days period.

Keyword: *Prediction, Artificial Neural Network, Support Vector Machine, Indeks Harga Saham Gabungan, Technical Analysis*

1 Pendahuluan

Saham merupakan instrumen pasar keuangan yang paling populer. Menerbitkan saham merupakan salah satu pilihan perusahaan ketika memutuskan untuk pendanaan perusahaan. Di sisi lain, saham merupakan instrumen investasi yang banyak dipilih para investor karena saham mampu memberikan keuntungan yang menarik [4], namun pergerakannya yang dinamis juga dapat menimbulkan kerugian sehingga investor harus berhati-hati dalam mengambil keputusan. Salah satu faktor yang bisa dipertimbangkan untuk pengambilan keputusan ini adalah indeks harga saham.

Indeks harga saham sendiri merupakan salah satu indikator utama pergerakan harga saham di suatu bursa saham [4]. Dengan membaca indeks dapat diketahui *trend* yang sedang terjadi di pasar, apakah sedang naik, turun, atau stabil sehingga investor dapat menentukan kapan untuk menjual, menahan atau membeli saham [15].

Berbagai cara bisa dilakukan oleh investor untuk mengetahui *trend* tersebut, diantaranya memantau langsung pergerakan saham saat itu juga atau menggunakan jasa analis saham dengan harapan dapat menterjemahkannya untuk mengambil keputusan selanjutnya dengan tepat. Investor juga dapat melakukan prediksi dengan melakukan analisa histori dan *trend* harga saham pada periode sebelumnya [1] sehingga penelitian dapat dilakukan untuk menyelesaikan permasalahan tersebut, mulai dari penelitian dari sisi analisis ekonomi hingga *machine learning*.

Machine learning sendiri adalah bagian dari teknologi kecerdasan artifisial yang lebih menitikberatkan pada konstruksi dan pelatihan sistem sehingga sistem ini dapat "belajar" dari data yang diberikan. Algoritma pembelajaran yang dimodelkan untuk memprediksi pergerakan indeks harga saham diantaranya adalah *Artificial Neural Network* (ANN) [1][14] dan *Support Vector Machine* (SVM) [6], dimana kedua metode tersebut

termasuk dalam *supervised learning* [7], dimana proses belajar yang dilakukan membutuhkan sesuatu yang memiliki pengetahuan tentang lingkungan.

2 Landasan Teori

2.1 Saham

Saham adalah tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan atau perusahaan terbatas. Wujud saham berupa selembar kertas yang menerangkan siapa pemiliknya [11]. Saham dijalankan di suatu lingkup yang dikenal dengan bursa saham. Bursa saham di Indonesia sendiri dijalankan oleh PT. Bursa Efek Indonesia (BEI) yang lebih dikenal dengan nama Indonesia Stock Exchange (IDX). Salah satu indeks harga saham yang sering menjadi acuan dalam IDX adalah Indeks Harga Saham Gabungan (IHSG). IHSG menggunakan semua emiten pada IDX sebagai komponen perhitungan indeks, yang berarti IHSG menggunakan semua harga saham yang tercatat di IDX karena masing-masing emiten mempunyai harga saham sendiri.

2.1.1 Analisis Teknikal Saham

Analisis teknikal didasarkan pada teori bahwa data historis mempengaruhi harga saham sekarang dan yang akan datang. Pola tersebut mengikuti pola *peak-and-through* (puncak dan lembah) yang amat sederhana tetapi efektif mengidentifikasikan pergerakan saham [11]. Hal tersebut tercermin pada pergerakan IHSG yang juga mempunyai periode naik dan turun. Analisis teknikal juga lebih bersifat untuk prediksi jangka pendek karena hanya menggunakan harga historis sebagai pedoman.

Indikator merupakan alat utama dari analisis teknikal. Untuk memberikan pedoman yang lebih akurat, digunakan beberapa indikator sekaligus, contohnya *moving average* indikator, *relative strength*, dan momentum [11]. Contoh indikator yang digunakan dapat dilihat pada tabel 1:

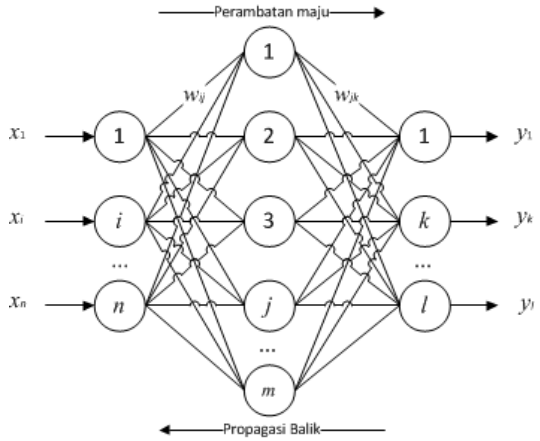
Tabel 1 Sepuluh indikator analisis teknikal saham [5]

No.	Nama Indikator	Formula
1	*Simple 10-day moving average*	$\frac{c_t + c_{t-1} + \cdots + c_{t-10}}{10}$
2	*Weighted 10-day moving average*	$\frac{((n) * c_t + (n-1) * c_{t-1} + \cdots + c_{t-10})}{(n + (n-1) + \cdots + 1)}$
3	Momentum	$C_t - C_{t-n}$
4	*Stochastic* K%	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} * 100$
5	*Stochastic* D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$
6	RSI (*Relative Strength Index*)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{t-i}(i,)/(\sum_{i=0}^{n-1} Dw_{t-i}/n)}$
7	MACD (*moving average convergence divergence*)	$MACD(n)_{t-1} + 2/n + 1 * (DIFF_t - MACD(n)_{t-1})$
8	*Larry William's* R%	$\frac{H_n - C_t}{H_n - L_n} * 100$
9	A/D *(Accumulation/Distribution) Oscillator*	$\frac{H_t - C_{t-1}}{H_t - L_t}$
10	CCI *(Commodity Channel Index)*	$\frac{M_t - SM_t}{0.015D_t}$

Keterangan dapat dilihat sebagai berikut :
C_t adalah harga penutupan pada waktu t, L_t adalah harga rendah, H_t adalah harga tinggi saat waktu t, DIFF: $\text{EMA}(12)_t - \text{EMA}(26)_t$, EMA (Exponential Moving Average), $\text{EMA}(k)_t$: $\text{EMA}(k)_{t-1} + \alpha \cdot C_t - \text{EMA}(k)_{t-1}$, α (*smoothing factor*) :2/1+*k*, *k* adalah periode waktu dari hari *k* dalam EMA, LL_t dan HH_t adalah harga terbawah dan tertinggi dalam beberapa hari t, M_t: $H_t + L_t + C_t/3$; SM_t: $(\sum_{i=1}^{n} M_{t-i+1}/n$, D_t: $(\sum_{i=1}^{n} |M_{t-i+1} - SM_t|)/n$, Up_t perubahan harga naik, D_w adalah perubahan harga turun pada waktu t.

2.2 *Artificial Neural Network* (ANN)

ANN adalah metode pembelajaran yang terinspirasi dari cara kerja sistem syaraf manusia, dimana terdapat *neuron* atau sel syaraf sebagai unit pemrosesan informasi yang merupakan dasar dari operasi. Terdapat banyak model ANN, namun dalam kasus ini digunakan model *Multi Layer Perceptron* (MLP) karena MLP sangat mudah dipelajari dan diimplementasikan sehingga banyak digunakan. Pelatihan MLP menggunakan algoritma pelatihan dimana salah satu yang populer adalah algoritma pelatihan *backpropagation*.



Gambar 1 Arsitektur ANN backpropagation

Perambatan maju dimulai dari perhitungan nilai *v* untuk setiap data masukan. *v* adalah hasil *x* dikalikan dengan *w* dimana *x* merupakan nilai fitur/vektor masukan, *w* adalah bobot vektor. Nilai *v* tersebut akan diaktivasi untuk menghasilkan sinyal keluaran dengan menggunakan fungsi aktivasi. Formulasi masing-masing fungsi aktivasi dapat dilihat pada tabel 2:

Tabel 2 Fungsi aktivasi untuk multi layer perceptron

No.	Fungsi Aktivasi	Formula
1	*Logistic Sigmoid*	$y = \frac{1}{1 + e^{-v}}$
2	*Tangent Transfer Sigmoid*	$y = \frac{2}{1 + e^{-v}} - 1$

Lakukan langkah perambatan maju hingga bertemu dengan sinyal *error* yang kemudian akan dirambatkan balik dari kanan ke kiri.

Sinyal *error* di *neuron* keluaran *k* pada *epoch* atau iterasi *p* diformulasikan:

$$e_k(p) = y_{dk}(p) - y_k(p) \quad (1)$$

$y_{dk}(p)$ adalah nilai keluaran yang menjadi target untuk *neuron k*, sedangkan $y_k(p)$ adalah keluaran aktual yang didapatkan oleh *neuron k* di *layer* keluaran.

Sinyal *error* yang didapatkan juga dihitung nilainya secara keseluruhan dengan *Mean Square Error* (MSE). MSE menandakan performansi dari proses pelatihan ANN yang dilakukan. Persamaan MSE adalah:

$$\text{MSE} = \frac{1}{p} \sum (e_k(p))^2 \quad (2)$$

Setelah didapatkan sinyal *error*, kemudian akan dilakukan propagasi balik. Karena sinyal balik pada semua *neuron* dalam *layer* keluaran disuplai langsung oleh nilai keluaran aktual, maka prosedur untuk memperbarui bobot w_{jk} yaitu:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta\, w_{jk}(p) \quad (3)$$

$\Delta\, w_{jk}(p)$ adalah koreksi bobot. Koreksi bobot dalam MLP dihitung dengan:

$$\Delta\, w_{jk}(p) = \eta \text{ x } y_j(p) \text{ x } \delta_k(p) \quad (4)$$

η adalah *learning rate*, sedangkan $\delta_k(p)$ adalah *gradient error* pada *neuron k* dalam layer keluaran pada iterasi ke- *p*.

Gradient error ditentukan dari turunan fungsi aktivasi terhadap v yang dikalikan dengan *error* pada *neuron* di layer keluaran, didapatkan:

$$\delta_k(p) = \frac{\partial y_k(p)}{\partial V_k(p)} \text{ x } e_k(p) \quad (5)$$

$y_k(p)$ adalah keluaran dari *neuron* k pada iterasi *p*, sedangkan $v_k(p)$ adalah akumulasi nilai/sinyal masuk terbobot yang masuk ke *neuron k* pada iterasi yang sama.

Pada peneletian, *untuk layer keluaran digunakan fungsi aktivasi logistic sigmoid sehingga pada contoh di bawah ini fungs*i aktivasi *logistic sigmoid* diturunkan terhadap $v_k(p)$. *Gradient error* yang didapatkan adalah:

$$\delta_k(p) = y_k(p) \text{ x } \left(1 - y_k(p)\right) \text{ x } e_k(p) \quad (6)$$

Untuk menghitung koreksi bobot pada *hidden layer*, cara yang sama juga dapat diterapkan:

$$\Delta\, w_{ij}(p) = \eta \text{ x } y_i(p) \text{ x } \delta_j(p) \quad (7)$$

$\delta_j(p)$ merepresentasikan *gradient error* pada *neuron j* dalam *hidden layer* dimana untuk fungsi aktivasi *tangent transfer sigmoid* diturunkan terhadap $v_j(p)$, sehingga berdasarkan notasi 5 yang digunakan untuk *neuron j*, *gradient error* yang didapatkan adalah:

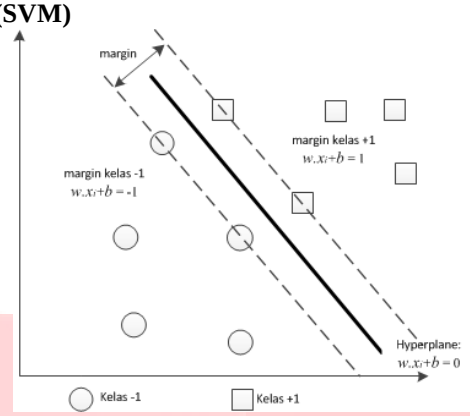
$$\delta_j(p) = \frac{1}{2} \text{ x } \left(1 - y_j(p)\right)^2 \text{ x } \sum_{k=l}^{l} \delta_k(p) \,.\, w_{jk}(p) \quad (8)$$

l adalah jumlah *neuron* pada layer keluaran. Setelah itu perbarui bobot pada *neuron hidden layer*:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta\, w_{ij}(p) \quad (9)$$

Propagasi balik selesai, kemudian lakukan perambatan maju dengan menggunakan *w* yang telah dikoreksi sebelumnya. Ulangi seluruh proses hingga mencapai kriteria MSE yang diinginkan atau batas iterasi/*epoch* yang diinginkan, mana yang tercapai terlebih dahulu.

2.3 *Support Vector Machine* (SVM)



Gambar 2Margin hyperplane untuk kasus linear

Diketahui suatu data *training* dengan format $(x_1, y_1), \dots, (x_i, y_i), \dots, (x_N, y_N)$, $x_i \in R^n, y_i \in \{+1, -1\}$ dengan $i = 1, 2, 3, \dots, N$, dimana N adalah banyaknya data. Diasumsikan kedua kelas dapat terpisah secara sempurna oleh *hyperplane* seperti pada gambar 2.

Margin adalah jarak antara *hyperplane* dengan data set terdekat dari masing-masing kelas yang akan menjadi acuan untuk pengklasifikasian. Data terdekat ini disebut sebagai *support vector*. Contoh *support vector* adalah data yang disimbolkan lingkaran dan bujursangkar yang dilewati tepat oleh garis putus-putus pada gambar 2

Margin antara dua kelas dapat dihitung dengan mencari jarak Secara spesifik, *margin* dihitung dengan cara sebagai berikut:

$$(w.x_i + b = 1) - (w.x_i + b = -1) \quad (10)$$

Margin hyperplane dibentuk oleh jarak antar dua *hyperplane* pada dua kelas tersebut, sehingga notasi 10 dapat diringkas menjadi:

$$\|w\| \text{ x } d = 2 \text{ atau } d = \frac{2}{\|w\|} \quad (11)$$

Permasalahan linear bisa dipisahkan dengan dengan suatu *hyperplane* dengan mudah, namun perlu diingat bahwa SVM mencari *hyperplane* dengan *margin* paling maksimal dengan memaksimalkan jarak antara *hyperplane* dengan data terdekat. Jarak ini telah dirumuskan dengan persamaan 2. Selanjutnya masalah ini diformulasikan ke dalam problem *Quadratic Programming* (QP) dengan cara meminimalkan *invers* persamaan 2, sehingga dapat dinotasikan:

$$min_w\ \tau(w) = \frac{1}{2} \parallel w \parallel^2 \quad (12)$$

dengan memperhatikan syarat:

$$y_i\,(w \cdot x_i + b) - 1 \geq 0,\ \forall i \quad (13)$$

Problem tersebut dapat dipecahkan dengan berbagai teknik komputasi, di antaranya dengan diubah ke dalam formula *lagrangian* yang menggunakan *Lagrange Multiplier*. Selanjutnya persamaan *Lagrange Multipliers* (*primal problem*) ditransformasikan ke bentuk *dual problem* yang lebih mudah diselesaikan, yaitu seperti berikut:

$$Ld = max_\alpha\, w\,(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \left(x_i \cdot x_j\right) \quad (14)$$

dengan syarat 1 yang berasal dari syarat optimasi dari *Lp* yang diturunkan terhadap *b*:

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad (15)$$

dan syarat 2 yang merupakan konstrain *Karush-Kuhn-Tucker* (KKT) untuk memaksimalkan *Lp* terhadap α_i:

$$\alpha_i \geq 0, i = 1,2, \dots, N \quad (16)$$

Perlu diketahui bahwa penjelasan sebelumnya berdasarkan asumsi bahwa kedua kelas dapat terpisah secara sempurna oleh *hyperplane*. Akan tetapi, pada kenyataannya kedua kelas tersebut tidak dapat terpisah secara sempurna. Hal ini menyebabkan proses optimisasi tidak dapat diselesaikan karena tidak ada *w* dan *b* yang memenuhi pertidaksamaan 2.18. Untuk itu, pertidaksamaan 4 dimodifikasi dengan memasukkan slack variable ξ_i ($\xi_i \geq 0$), menjadi:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (17)$$

Demikian juga untuk persamaan 3 dimodifikasi menjadi:

$$\frac{1}{2} \parallel w \parallel^2 + C \sum_{i=0}^{n} \xi_i \quad (18)$$

Parameter C berguna untuk mengontrol *trade-off* antara *margin* dan *error* klasifikasi. Semakin besar nilai C, maka semakin besar pula pinalti yang dikenakan untuk tiap klasifikasi.

Data yang ada dalam kehidupan nyata, seperti data saham tidak bisa diklasifikasikan secara sempurna menggunakan SVM secara *linear*. Pada perrmasalahan *non-linear*, maka pendekatan yang digunakan adalah memetakan data inputan ke ruang fitur dimensi yang lebih tinggi, dimana SVM memerlukan kernel sebagai pengganti $(x_i \cdot x_j)$ pada persamaan 14 yang dinotasikan sebagai $\varphi(x_i).\varphi(x_j)$. Nilai dua *dot-product* kedua buah vektor ini dapat dihitung secara tidak langsung, yaitu tanpa mengetahui fungsi transformasi φ. Teknik komputasi ini kemudian disebut sebagai *kernel trick*:

$$K(x_i, x_j) = \varphi(x_i).\varphi(x_j) \quad (19)$$

Dan untuk prediksi pada set data dengan dimensi fitur yang baru diformulasikan:

$$f(\varphi(\mathrm{x})) = sign\ (w.\varphi(\mathrm{z}) + b = sign(\textstyle\sum_{i=1}^{N} \alpha_i\, y_i\, \varphi(x_i).\varphi(\mathrm{z}) + b \quad (20)$$

N adalah jumlah data yang menjadi *support vector*, x_i adalah *support vector*, dan z adalah data uji yang akan dilakukan prediksi.

Tabel 3Fungsi kernel pada SVM non-linear [3]

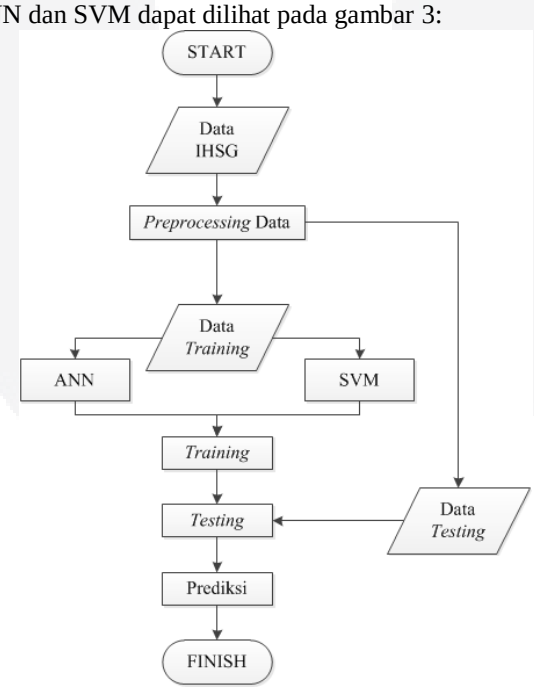
No.	Fungsi Kernel	Formula
1	*Polynomial*	$K(x_i, x_j) = (\gamma(x_i.x_j) + r)^d$
2	*Gaussian Radial Basis (RBF)*	$K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2)$
3	*Sigmoid*	$K(x_i, x_j) = \tanh(\gamma(x_i.x_j) + r)$

Dengan parameter γ, r, d > 0.

3 Perancangan Sistem

3.1 Deskripsi Sistem

Sistem yang akan diimplementasikan dalam tugas akhir ini mempunyai tujuan utama untuk memprediksi pergerakan indeks harga saham yang diklasifikasikan menjadi pergerakan naik atau pergerakan turun. Hasil keluaran dari sistem akan dihitung akurasinya dengan cara membandingkannya dengan data pergerakan indeks harga saham sebenarnya yang telah melalui *preprocessing* terlebih dahulu sehingga data yang menjadi target dari sistem dapat dibandingkan dengan hasil keluaran sistem. Tahapan yang akan dilakukan untuk membangun sistem prediksi, baik untuk ANN dan SVM dapat dilihat pada gambar 3:



Gambar 3 Alur proses prediksi pergerakan indeks harga saham menggunakan ANN dan SVM

3.2 Perancangan Sistem

3.2.1 Perancangan Data

Data yang digunakan untuk memprediksi pergerakan indeks harga saham merupakan data harian IHSG [16]. Data tersebut dibagi menjadi dua bagian menjadi data latih (*training*) dan data uji (*testing*) yang bisa dilihat pada tabel 4:

Tabel 4 Komposisi data yang digunakan

Jenis Komposisi	Data Latih	Data Uji
Rentang Waktu	Januari 2011 – Desember 2012	Januari 2013 – Desember 2014
Jumlah hari transaksi	491 hari	484 hari
Persentase	50,4 %	49,6 %

3.2.2 *Preprocessing*

1. Tersedia data indeks harga saham, dalam hal ini data indeks harga saham yang digunakan adalah data harian IHSG [16].
2. Untuk masukan sistem, dilakukan proses perhitungan indikator teknikal yang telah ditentukan dengan menggunakan data IHSG sebagai sumber perhitungan.
3. Bagian dari data harian IHSG yang digunakan sebagai target dari sistem adalah atribut *close*, dimana atribut *close* ini digunakan sebagai pedoman pergerakan indeks harga saham. Atribut *close* menghasilkan kelas target yang diklasifikasikan menjadi dua kelas yang berbeda. Sebelum dilakukan pengklasifikasian, terlebih dahulu dilakukan perhitungan:

$$x = x_t - x_{t-1} \tag{21}$$

keterangan:

x : selisih antara dua rentang nilai *close*

x_t : nilai *close* pada transaksi hari ke-t

x_{t-1} : nilai *close* pada transaksi hari sebelumnya (t-1)

Untuk pengklasifikasian menjadi kelas target (T), syarat yang digunakan adalah sebagai berikut:

$$T = sign(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases} \tag{22}$$

4. Buat *input* masukan dan target untuk ANN dan SVM. Masukan sendiri disesuaikan dengan skenario pengujian yang digunakan.

4 Analisis dan Pengujian

Skenario pengujian dibagi menjadi tiga bagian dengan penjelasan sebagai berikut:

1. Tujuan pengujian skenario 1: mendapatkan hasil berupa parameter ANN yang terbaik dengan *input* berupa komposisi analisis teknikal dengan perhitungan periode waktu 10 hari transaksi.

Tabel 5 Kombinasi Parameter sistem untuk pengujian ANN

Parameter ANN	Level
jumlah *neuron* pada *hidden layer*	1, 3[12], 10, 50, 100, 500, 1000
epoch	1000
η (learning rate)	0.1 [12]

2. Tujuan pengujian skenario 2: mendapatkan hasil berupa parameter SVM yang terbaik dengan *input* berupa komposisi analisis teknikal dengan perhitungan periode waktu 10 hari transaksi.

Tabel 6 Kombinasi Parameter sistem untuk pengujian SVM

Parameter SVM	Tipe Kernel
	Polynomial
d (degree)	1, 2, 3[3], 4, 5
γ (gamma)	2^{-8}, 2^{-7}, …, 2^{8}[3]
C	2^{-15}, 2^{-14}, …, 2^{12}[3]

3. Skenario 3 bertujuan untuk mengukur performansi mana yang lebih baik dari kedua metode dengan membandingkan akurasinya dengan melakukan pengujian terhadap *input* analisis teknikal dengan perhitungan periode waktu yang berbeda-beda. Parameter ANN dan SVM yang digunakan berasal dari hasil pengujian terhadap dua skenario sebelumnya.

Tabel 7 Berbagai input masukan untuk pengujian skenario 3

Studi Kasus (input)	Periode waktu (n) yang digunakan
input 1	3 hari transaksi
input 2	5 hari transaksi
input 3	10 hari transaksi

4.1 Hasil dan Analisis Pengujian Skenario 1

Pengujian parameter *hidden neuron* ANN dapat dilihat pada tabel berikut:

Tabel 8 Hasil pelatihan dan pengujian hidden neuron dengan epoch = 1000 dan η = 0,1

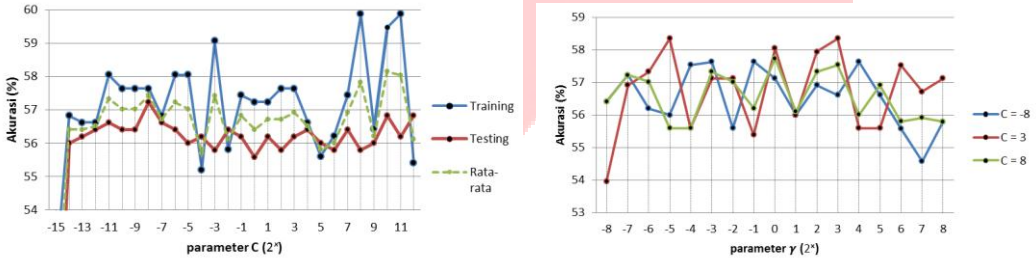
Hidden Neuron	**Rata-rata *training***	**Rata-rata *testing***	***Average***	**Rata-rata MSE**
3	55.51934	56.59092	56.05513	0.243904
10	55.62117	56.46695	56.04406	0.243746
50	54.501	54.97936	54.74018	0.24238
100	55.51889	55.57851	55.5487	0.241651
500	56.78207	55.55785	56.16996	0.228399

Dari tabel 8 terlihat bahwa pada saat *hidden neuron* berjumlah 500, maka *hidden neuron* mempunyai rata-rata akurasi training terbaik yaitu sebesar 56,17%. Pada tabel diatas juga terlihat bahwa *hidden neuron* yang berjumlah di atas 100 mempunyai akurasi *training* yang lebih baik, namun akurasi *testing* yang didapatkan semakin rendah.

Parameter ANN terbaik yang didapatkan pada skenario 1: *epoch* sebesar 1000, η = 0,1 dan 3 *hidden neuron*.

4.2 Hasil dan Analisis Pengujian Skenario 2

Pencarian parameter C dan γ terbaik menggunakan jarak antar nilai sebesar 2^x serta menggunakan kombinasi parameter d = 3 berdasarkan penelitian Lin [3]. Hasil pengujian untuk lebih jelasnya dapat dilihat pada gambar 4:

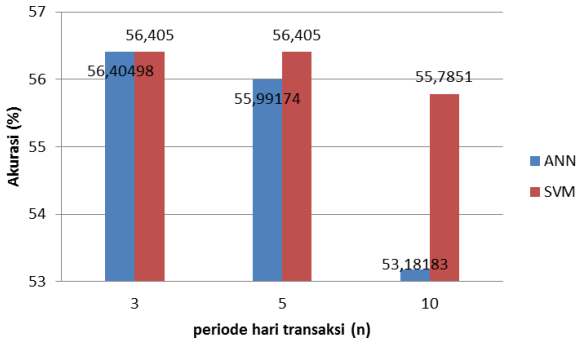


Gambar 4 Grafik parameter C dan γ yang diujikan

Dari gambar 4 (kiri) dapat disimpulkan bahwa semakin besar parameter C yang diuji, maka semakin besar pula potensi data *training* mengalami *overfit*, ditandai dengan akurasi data *training* yang lebih besar daripada akurasi yang didapatkan data *testing*. Dari gambar 4 (kanan) dapat disimpulkan bahwa semakin besar nilai parameter γ yang diujikan, maka akurasi yang didapatkan semakin rendah

Parameter SVM terbaik yang didapatkan pada skenario 2: parameter C = 8, γ = 0 dan d = 3.

4.3 Hasil dan Analisis Pengujian Skenario 3



Gambar 5 Perbandingan hasil akurasi prediksi ANN dan SVM dengan input analisis teknikal saham per periode waktu (n)

Dapat dilihat pada gambar 5 bahwa SVM mengungguli ANN dari segi akurasi prediksi *trend* pergerakan indeks harga saham IHSG, walaupun ANN bisa mengimbangi akurasi SVM pada hasil prediksi dengan input analisis teknikal saham pada periode waktu 3 hari. Rata-rata semua model semakin menurun seiring dengan pengujian menggunakan *input* indikator teknikal dengan periode waktu yang lebih lama, namun secara keseluruhan untuk setiap *input* periode waktu yang diujikan, penurunan akurasi SVM lebih stabil jika dibandingkan dengan akurasi ANN.

Penggunaan model SVM dan ANN untuk memprediksi pergerakan indeks harga saham, khususnya dalam penggunaan sepuluh analisis indikator teknikal pada data IHSG cukup cocok dengan hasil akurasi yang cukup baik dan kompetitif antar model apabila perhitungan periode waktu yang digunakan untuk jangka waktu pendek

seperti tiga hari. Penggunaan input indikator teknikal pada data IHSG untuk periode waktu yang lebih panjang menyebabkan akurasi prediksi antar model semakin menurun dan tidak kompetitif.

5 Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang bisa diambil dalam tugas akhir yang dilakukan adalah:

1. Dari hasil penelitian yang dilakukan, SVM mengungguli ANN dari segi akurasi prediksi pergerakan harga saham IHSG dengan akurasi tertinggi SVM sebesar 56,405% dan akurasi tertinggi ANN sebesar 56,40498% untuk data masukan berupa analisis teknikal saham dengan periode waktu tiga hari.
2. Semakin besar nilai parameter *hidden neuron* pada ANN, maka potensi data *training* ANN mengalami *overfit* semakin besar.
3. Semakin besar nilai parameter C dan parameter γ pada SVM, maka potensi data *training* SVM mengalami *overfit* semakin besar.
4. Data masukan sistem berupa analisis teknikal saham dengan periode waktu lebih pendek (3 hari) mempunyai tingkat akurasi prediksi yang lebih baik dibandingkan data masukan analisis teknikal saham dengan periode waktu yang lebih lama (5 hari dan 10 hari) baik untuk ANN maupun SVM.

5.2 Saran

Saran yang ingin disampaikan untuk pengembangan lebih lanjut adalah:

1. Untuk penelitian selanjutnya, khususnya untuk model *neural network* agar digunakan parameter tambahan, seperti penggunaan momentum atau *varian* dari *neural network* lainnya.
2. Untuk penelitian dengan menggunakan data masukan analisis teknikal saham sebaiknya menggunakan perhitungan periode waktu 3 hari.
3. Perlu penelitian lebih lanjut untuk mengetahui pengaruh indikator analisis teknikal saham yang digunakan terhadap model prediksi

DAFTAR PUSTAKA

[1] Afrianto, R., B., Handayani, T., Arieshanti, I. 2014. Prediksi Pergerakan Harga Saham Menggunakan Metode Backpropagation Neural Network. *Jurnal Simantec*, 3(3), ISSN 2088-2130.

[2] Artha, D., R., Achsani, N., A., & Sasongko. 2014. Analisis Fundamental, Teknikal, dan Makroekonomi Harga Saham Sektor Pertanian. *Jurnal Manajemen dan Kewirausahaan*, 16(2), 175-184.

[3] Chang, C. C., & Lin, C. J. 2001. *LIBSVM: A library for support vector machines*. Taipei, Taiwan: National Taiwan University. <http://www.csie.ntu.edu.tw/?cjlin/libsvm> (version 3.20).

[4] Indonesia Stock Exchange . 2010. Buku Panduan Indeks Harga Saham Bursa Efek Indonesia. Jakarta: PT. Bursa Efek Indonesia. <http://www.idx.co.id/ >

[5] Kara, Y., Boyacioglu, M. A., & Baykan, Ö., K. 2011. Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, 38(5), 5311–5319.

[6] Kim, K., J. 2003. Financial Time Series Using Support Vector Machines. *Neurocomputing*, 55, 307-319.

[7] Kotsiantis, S., B. 2007. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31, 249-268.

[8] Rusu, V., Rusu, C. 2003. Forecasting Methods and Stock Market Analysis. *Creative Math*, 12, 103-110.

[9] Stockcharts.com. *Technical Indicators and Overlays*. Diperoleh dari http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators (diakses tanggal 30 Mei 2015)

[10] Sarle, Warren S. *Ill Conditioning in Neural Networks*. Cary, NC, USA: SAS Institute Inc. Diperoleh dari ftp://ftp/sas .com/pub/neural/illcond/illcond.html (diakses tanggal 31 Juli 2015)

[11] Sukamulja, S. 2005. Analisis Fundamental, Teknikal, dan Program Metastock. *Finance Club Training*. Yogyakarta: Universitas Gajah Mada.

[12] Suyanto. 2007. *Artificial Intelligence: Searching, Reasoning, Planning, dan Learning*. Bandung: Informatika Bandung.

[13] Timor, M., Dinçer, H., & Emir, Ş. 2012. A Stock Selection Model Based on Fundamental and Technical Analysis Variables by Using Artificial Neural Networks and Support Vector Machines. *Review of Economics & Finance*, 2, 106-122.

[14] Tiphimmala, S. 2014. Forecasting Stock Price Index Using Artificial Neural Networks in the Indonesian Stock Exchange. *Thesis*. Yogyakarta: Universitas Atma Jaya Yogyakarta.

[15] Utami, A., S. 2014. Indeks Harga Saham Gabungan. *Makalah*. Jember: Universitas Muhammadiyah Jember.

[16] Yahoo! Finance. *JKSE Historical Prices*. Diperoleh dari http://finance.yahoo.com/q/hp?s=%5EJKSE+Historical+Prices (diakses tanggal 20 Mei 2015)