

Prediksi Harga Saham Menggunakan *Hidden Markov Model* (HMM) dan *Fuzzy Model*

Devy Yendriani¹, Jondri, M.Si.², Untari Novia Wisesty, MT³

Fakultas Informatika Telkom University, Bandung

Abstrak

Saham merupakan salah satu pilihan investasi yang menarik karena dapat diperoleh untung yang besar. Agar investor mendapatkan untung dan menghindari rugi, diperlukan perhatian yg jeli terhadap pergerakan saham. Salah satu caranya adalah dengan memprediksi saham.

Pada tugas akhir ini, teknik yang diusulkan adalah dengan menggunakan *Hidden Markov Model* (HMM) dan *Fuzzy Logic* (FL). HMM sudah sering digunakan untuk menganalisa dan memprediksi fenomena *time series*. HMM digunakan untuk mengelompokkan data yang mempunyai kemiripan *data pattern*. FL digunakan untuk memprediksi harga saham dengan *Gradient Descent* untuk mengoptimasi aturan *fuzzy* yg dibentuk.

Sistem prediksi ini menggunakan data saham harian PT Telekomunikasi Indonesia, Tbk dari tanggal 3 Januari 2011 hingga 20 Desember 2014. Sistem ini memiliki error MAPE sebesar 1.21.

Kata Kunci: *time series*, harga saham, *Hidden Markov Model*, *Fuzzy Logic*, *Gradient Descent*.

Abstract

Stock is an interesting investment option because it can generate a big profit. In order for investors to make profits and avoid losses, an attention to their movement is necessary. One of them is to predict the stock.

In this thesis, the proposed technique is Hidden Markov Model (HMM) and Fuzzy Logic (FL). HMM has been frequently used to analyze and predict the phenomenon of time series. HMM is used to classify datas that has some similarities in data pattern. FL is used to predict stock prices with Gradient Descent to optimize fuzzy rules that formed.

The prediction system uses daily stock data of PT Telekomunikasi Indonesia from January 3rd, 2011 to December 20th, 2014. This system has a MAPE error of 1.21.

Keywords: *time series*, stock price, *Hidden Markov Model*, *Fuzzy Logic*, *Gradient Descent*.

I. Pendahuluan

1.1 Latar Belakang

Saham adalah surat berharga yang menyatakan penyertaan atau pemilihan seseorang atau badan dalam suatu perusahaan.^[1] Saham merupakan salah satu pilihan investasi yang menarik karena dapat diperoleh untung yang besar.^[4] Selama 60 tahun terakhir, telah banyak yang meneliti secara intensif terhadap bursa saham dengan harapan dapat mengalahkan pasar. Mengalahkan pasar dapat diartikan dengan memiliki return rate yang lebih tinggi daripada return rate pasar secara konsisten sekaligus menjaga tingkat risiko yang sama dengan pasar.^[9]

Salah satu penelitian yang banyak dilakukan adalah memprediksi harga saham. Beberapa teknik *Artificial Intelligence* (AI) seperti jaringan saraf tiruan (JST), *fuzzy logic* (FL), dan *Support Vector Machines* (SVMs) telah diusulkan untuk mendapatkan hasil prediksi yang lebih baik.

Pada tugas akhir ini, penulis akan menggunakan *Hidden Markov Model* (HMM) dan *Fuzzy Logic* (FL) untuk memprediksi harga saham PT Telekomunikasi Indonesia, Tbk. *Hidden Markov Model* adalah salah satu tipe model Markov dimana *states* nya tidak tampak. Metode ini sudah sering digunakan untuk menganalisa dan memprediksi fenomena *time series*. Model ini sudah digunakan di area seperti *speech recognition*, *DNA sequencing*, dan analisis *ECG*.^[4] *Fuzzy Logic* populer dalam penelitian yang berhadapan dengan sistem non-linear. Pada penelitian ini HMM digunakan untuk mengidentifikasi *data pattern* yang mirip dan

mengelompokkannya berdasarkan kemiripannya tersebut ke dalam *bucket*. Data-data yang telah *bucket*-kan kemudian akan diprediksi menggunakan FL. Pada proses prediksi, FL akan dibantu oleh *Gradient Descent* dalam mengoptimasi aturan *fuzzy*.

1.2 Tujuan

Tujuan dari tugas akhir ini adalah:

1. Merancang dan mengimplementasikan sistem prediksi harga saham dengan menggunakan metode *Hidden Markov Model* dan *Fuzzy Logic*.
2. Menganalisis performansi sistem prediksi harga saham yang dihasilkan.

2. Landasan Teori

2.1 Hidden Markov Model

Hidden Markov Model (HMM) adalah sebuah model statistik perkembangan dari rantai Markov dimana keadaan yang akan datang dari suatu *sequence* tidak hanya ditentukan oleh keadaan saat ini, tetapi juga perpindahan dari suatu *state* ke *state* yang lain. Model Markov pada umumnya memiliki *state* yang langsung dapat diamati. Namun pada HMM *state*-nya tidak dapat diamati secara langsung, akan tetapi yang dapat diamati adalah variabel-variabel yang terpengaruh oleh *state* (observasi).

Salah satu contoh dari *Hidden Markov* ini adalah perkiraan cuaca. Seseorang yang berada didalam sebuah ruangan tertutup diminta untuk menebak cuaca hari ini. Misalkan keadaan cuaca ada 3: hujan, berawan, dan

panas. Orang tersebut tidak dapat keluar untuk melihat keadaan cuaca. Namun dia dapat memperhatikan orang yang masuk ke dalam ruangan tersebut, apakah membawa payung atau tidak.

1. Elemen HMM

Sebuah HMM mempunyai elemen-elemen sebagai berikut:

1. N, yaitu jumlah state, dengan ruang state $S = \{s_1, s_2, \dots, s_N\}$ dan state pada waktu t dinyatakan dengan q_t .
2. M, yaitu jumlah pengamatan (observasi) tiap state, dengan ruang observasi $V = \{v_1, v_2, \dots, v_M\}$
3. $A = [a_{ij}]$, yaitu matriks peluang transisi dimana a_{ij} menyatakan probabilitas transisi dari state i ke state j.
4. $B = [b_{im}]$, yaitu matriks peluang bersyarat observasi v_k pada state j.
5. π_i yaitu distribusi state awal.

dimana O merupakan barisan observasi, misal O_1, O_2, \dots, O_T . Q barisan state q_1, q_2, \dots, q_T pada model Markov dan T merupakan panjang / lama observasi. Dari elemen diatas, Hidden Markov Model dapat dituliskan dalam

notasi $\lambda = (A, B, \pi)$.

2. Tiga Masalah Umum HMM

Agar HMM dapat diaplikasikan ke berbagai masalah nyata, ada tiga masalah mendasar dalam HMM yang harus diselesaikan, yakni:

3. Evaluation problem

Diberikan model $\lambda = (A, B, \pi)$, bagaimana mencari $P(O|\lambda)$, peluang dari barisan observasi $O = O_1, O_2, \dots, O_T$? Permasalahan ini dapat diselesaikan dengan menggunakan algoritma forward-backward.

4. Decoding problem

Diberikan barisan observasi $O = O_1, O_2, \dots, O_T$ dan model λ , bagaimana memilih $Q = q_1, q_2, \dots, q_T$

yang paling optimal? Untuk menyelesaikan permasalahan ini, digunakan algoritma Viterbi

5. Learning problem

Bagaimana mengatur parameter model $\lambda = (A, B, \pi)$ agar $P(O|\lambda)$ maksimum? Untuk menyelesaikan masalah ini digunakan Algoritma Baum-Welch.

$$P(O|\mu, \Sigma) = \frac{1}{(2\pi)^n |\Sigma|} \exp\left(-\frac{1}{2} (O - \mu)' \Sigma^{-1} (O - \mu)\right)$$

Untuk mendapatkan HMM mean dan variansi dari GMM diperlukan variabel δ yang diperoleh berdasarkan:

$$\delta_i(t) = \sum_{m=1}^M \alpha_j(t-1) \alpha_{ji} w_{im} b_{im}(o_t) \beta_i(t)$$

sebagai variabel a sebelum a M adalah total komponen pada umny

GMM pada saat state i dan N adalah jumlah state. Selain itu dibutuhkan variabel untuk menunjukkan kemungkinan keberadaan di state j pada waktu t dengan komponen ke-k campuran pada saat pengamatan o_t :

$$\gamma_{ij}^k(t) = \frac{\alpha_j(t-1) \alpha_{ji} w_{im} b_{im}(o_t) \beta_i(t)}{\delta_i(t)}$$

sehingga diperoleh formula untuk re-estimasi parameter sebagai berikut:

$$\mu'_i = \frac{\sum_{t=1}^T \sum_{j=1}^N \gamma_{ij}^k(t) o_t}{\sum_{t=1}^T \sum_{j=1}^N \gamma_{ij}^k(t)}$$

$$\Sigma'_i = \frac{\sum_{t=1}^T \sum_{j=1}^N \gamma_{ij}^k(t) (o_t - \mu'_i)(o_t - \mu'_i)'}{\sum_{t=1}^T \sum_{j=1}^N \gamma_{ij}^k(t)}$$

$$\pi'_i = \frac{\sum_{t=1}^T \gamma_{ij}^k(t)}{\sum_{t=1}^T \sum_{j=1}^N \gamma_{ij}^k(t)}$$

2.2 Fuzzy Logic

Logika Fuzzy adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian.^[8] Logika fuzzy memiliki nilai keaburan atau [6]

kesamaran antara benar dan salah. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam

istilah biner (0 atau 1, hitam atau putih, ya atau tidak), logika fuzzy menggantikan kebenaran boolean dengan tingkat kebenaran.^[8]

Logika Fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keaburan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat".

Continuous Density HMM

Pada sub-bab sebelumnya, telah dibahas HMM yang berhubungan dengan urutan simbol diskrit. Sedangkan yang akan digunakan pada tugas akhir ini adalah HMM yang dapat mengatasi *dataset* kontinu. Digunakan pendekatan kepadatan pengamatan secara terus menerus yaitu *probability density function* (pdf). Prosedur re-estimasi yang merepresentasikan pdf adalah: ^[10]

$$p(\mathbf{O}) = \sum_{m=1}^M \alpha_m \prod_{i=1}^N N(\mathbf{O}_i; \mu_{im}, \Sigma_{im})$$

α_i adalah vector pengamatan, w_{im} adalah bobot pada Gaussian *mixture*, dimana w harus memenuhi batasan sebagai berikut:

$$\sum_{m=1}^M \alpha_m = 1, \quad 1 \leq i \leq N$$

$$w_{im} \geq 0, \quad 1 \leq i \leq N, 1 \leq m \leq M$$

$N(\mathbf{O}_i; \mu_{im}, \Sigma_{im})$ adalah multivarian GMM dengan *mean vector* μ' dan matrik kovarian Σ' :

1. Himpunan *fuzzy* dan fungsi keanggotaan
 Dalam teori logika *fuzzy* dikenal himpunan *fuzzy* yang merupakan pengelompokan sesuatu berdasarkan variabel bahasa yang dinyatakan dalam fungsi keanggotaan. Di dalam *universe of discourse* U, fungsi keanggotaan dari suatu himpunan *fuzzy* tersebut bernilai antara 0.0 sampai dengan 1.0.^[6] Contohnya adalah himpunan dari suhu atau temperatur, dapat dinyatakan dengan dingin, hangat, panas.

Tabel 2-1 *Fuzzy set* untuk suhu

Suhu (° C)	Dingin	Hangat	Panas
5	1	0,1	0
15	0,9	0,8	0
25	0,5	1	0,6
35	0,1	0,6	0,9
45	0	0,2	1

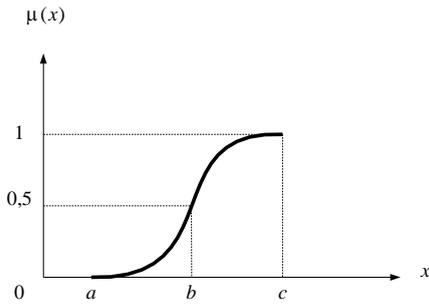
Tabel diatas menunjukkan beberapa temperatur dan derajat keanggotaan masing-masing temperature untuk

dingin, hangat, dan panas. Suhu 5° memiliki nilai penuh untuk derajat keanggotaan dingin, 0,1 untuk derajat keanggotaan hangat, dan 0 untuk derajat keanggotaan panas.

Fungsi keanggotaan dari suatu himpunan fuzzy dinyatakan dengan derajat keanggotaan. Fungsi keanggotaan dapat dinyatakan dengan berbagai bentuk, yaitu:

1. Fungsi sigmoid

dalam interval [0,1]. Grafik notasi matematika untuk fungsi ini adalah sebagai berikut:

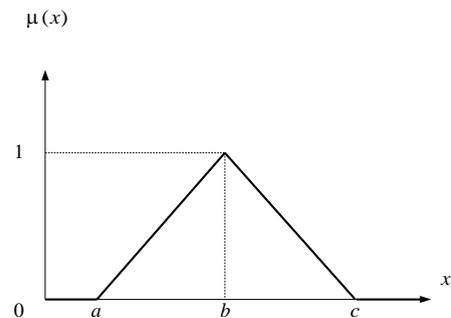


Gambar 2-1 Grafik fungsi sigmoid

$$\text{Sigmoid}(x, a, b, c) = \begin{cases} 0, & x \leq a \\ 2((x-a)/(c-a))^2, & a < x \leq b \\ 1 - 2((c-x)/(c-a))^2, & b < x < c \\ 1, & c \leq x \end{cases}$$

2. Fungsi segitiga

Pada fungsi ini hanya terdapat satu nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika x = b. tetapi nilai-nilai disekitar b memiliki derajat keanggotaan yang turun cukup tajam. Grafik dan notasi matematika dari fungsi ini adalah sebagai berikut:

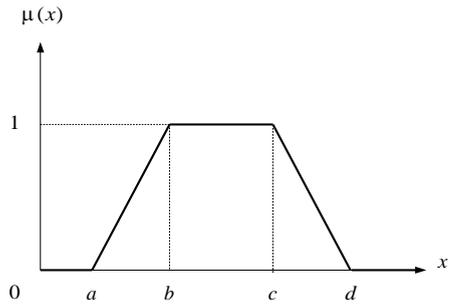


Gambar 2-2 Grafik fungsi segitiga

$$\text{Segitiga}(x, a, b, c) = \begin{cases} 0, & x \leq a, x \geq c \\ (x-a)/(b-a), & a < x \leq b \\ -(x-c)/(c-b), & b < x \leq c \end{cases}$$

3. Fungsi trapezium

Pada fungsi ini terdapat beberapa nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika b ≤ x ≤ c. Tetapi derajat keanggotaan untuk a < x < b dan c < x < d memiliki karakteristik yang sama dengan fungsi segitiga. Grafik dan notasi matematika dari fungsi trapezium sebagai berikut:

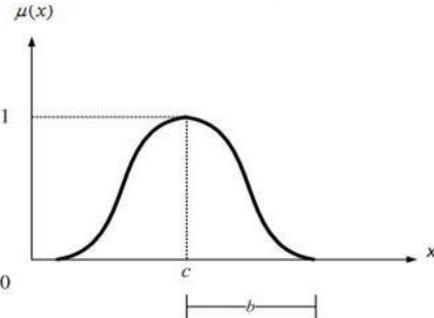


Gambar 2-3 Grafik fungsi trapezium

$$\text{Trapezium}(x, a, b, c, d) = \begin{cases} 0, & x \leq a, x \geq d \\ (x-a)/(b-a), & a < x < b \\ 1, & b \leq x \leq c \\ -(x-d)/(d-c), & c < x < d \end{cases}$$

4. Fungsi Gaussian

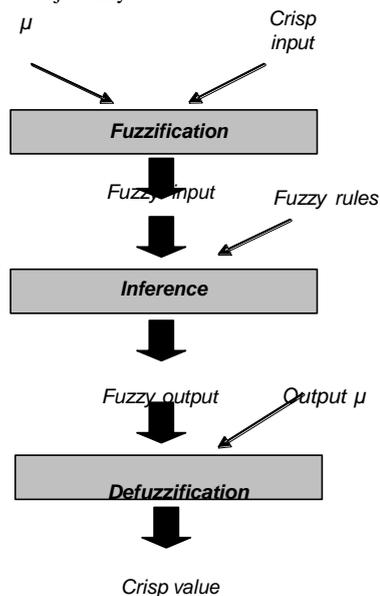
Fungsi keanggotaan Gaussian ditentukan oleh parameter c dan b yang merupakan titik tengah dan lebar fungsi. Grafik dan notasi matematika dari fungsi tersebut adalah sebagai berikut:



Gambar 2-4 Grafik fungsi Gaussian

Gauss $\diamond \diamond \diamond \diamond = \diamond \diamond \diamond \diamond$

2. Cara kerja fuzzy



Gambar 2-5 Diagram blok untuk sistem berbasis aturan fuzzy

a. *Fuzzification*

Fuzzification merupakan proses mengubah masukan-masukan yang nilai kebenarannya bersifat pasti (*crisp input*) ke dalam bentuk *fuzzy input*, yang berupa nilai linguistik yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu. Untuk mengubah *crisp input* menjadi *fuzzy input*, terlebih dahulu harus menentukan *membership function* untuk tiap *crisp input*, kemudian proses *fuzzification* akan mengambil *crisp input* dan membandingkan dengan *membership function* yang telah ada untuk menghasilkan nilai *fuzzy input*.

b. *Inference*

Inference melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Proses *inference* memperhitungkan semua aturan yang ada dalam basis pengetahuan. Secara sintaks, suatu aturan *fuzzy* dituliskan sebagai:

IF antecedent THEN consequent.

Dalam suatu sistem berbasis aturan *fuzzy*, proses *inference* memperhitungkan semua aturan yang ada dalam basis pengetahuan, hasil dari proses *inference* direpresentasikan oleh suatu *fuzzy set* untuk setiap variabel bebas (pada *consequent*). Derajat keanggotaan untuk setiap nilai variabel tidak bebas menyatakan ukuran kompatibilitas terhadap variabel bebas (pada *antecedent*). Misalkan terdapat suatu sistem dengan n variabel bebas x1, ..., xn dan m variabel tidak bebas y1, ..., ym. Misalkan R adalah suatu basis dari sejumlah r aturan fuzzy

IF P1(x1,...,xn) THEN Q1(y1,...,ym),
IF Pr(x1,...,xn) THEN Qr(y1,...,ym),

dimana P1, ... Pr menyatakan *fuzzy predicate* untuk variabel bebas, dan Q1, ... Qr menyatakan *fuzzy predicate* untuk variabel tidak bebas.

Terdapat dua model aturan *fuzzy* yang digunakan secara luas dalam berbagai aplikasi, yaitu:

a. Model Mamdani

Pada model ini, aturan *fuzzy* didefinisikan sebagai:

IF x1 is A1 AND ...AND xn is An THEN y is B,

Dimana A1, ..., An, dan B adalah nilai-nilai linguistik (atau fuzzy set) dan "x1 is A1" menyatakan bahwa nilai variabel x1 adalah anggota fuzzy set A1.

b. Model Takagi Sugeno

Model ini menggunakan aturan yang berbentuk:

IF x1 is A1 AND ...AND xn is An THEN y = f(x1, ..., xn),

Dimana f bisa berupa sembarang fungsi dari variabel - variabel input yang nilainya berada dalam interval variabel output. Biasanya, fungsi ini dibatasi dengan menyatakan f sebagai kombinasi linier dari variabel - variabel input:

$$f(x1, \dots, xn) = w0 + w1x1 + \dots + wnxn,$$

dimana w0, w1, ..., wn adalah konstanta yang berupa bilangan real yang merupakan bagian dari spesifikasi aturan fuzzy.

c. *Defuzzification*

Defuzzification merupakan proses mengubah *fuzzy output* menjadi *crisp value* berdasarkan fungsi keanggotaan yang telah ditentukan. Proses ini merupakan kebalikan dari proses *fuzzification*. Terdapat berbagai metode *defuzzification*, yaitu:

a. *Centroid method*

Metode ini juga disebut pusat area. Metode ini lazim dipakai dalam proses *defuzzification*. Metode ini menghitung nilai *crisp* menggunakan rumus:

$$z = \frac{\sum_{i=1}^n \mu_i \cdot z_i}{\sum_{i=1}^n \mu_i}$$

b. *Height method*

Metode ini dikenal juga sebagai prinsip keanggotaan maksimum karena metode ini secara sederhana memilih nilai *crisp* yang memiliki derajat keanggotaan maksimum. Metode ini hanya bisa dipakai untuk fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada semua nilai *crisp* lainnya.

c. *First (or Last) of Maxima*

Metode ini merupakan generalisasi dari *height method* untuk kasus dimana fungsi keanggotaan *output* memiliki lebih dari satu nilai maksimum. Sehingga nilai *crisp* yang digunakan adalah salah satu dari nilai yang dihasilkan dari maksimum pertama atau maksimum terakhir.

d. *Mean-Max method*

Metode ini merupakan generalisasi dari *height method* untuk kasus dimana terdapat lebih dari satu nilai *crisp* yang memiliki derajat keanggotaan maksimum. Sehingga y* didefinisikan sebagai titik tengah antara nilai *crisp* terkecil dan nilai *crisp* terbesar:

$$z = \frac{m + \text{max}}{2}$$

e. *Weighted average*

Metode ini mengambil nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan. Sehingga y* didefinisikan sebagai:

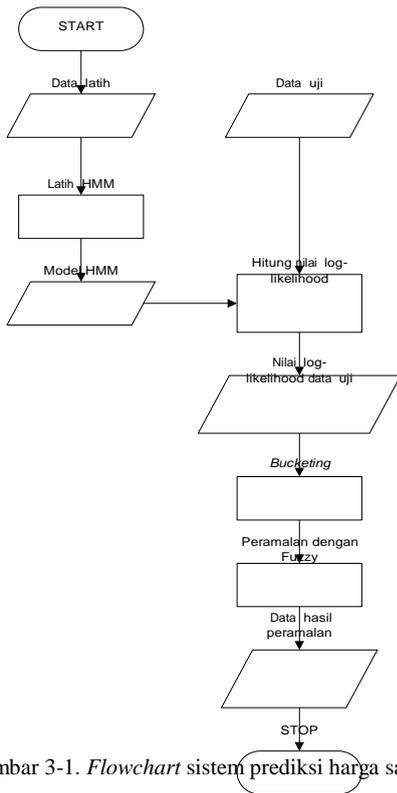
$$z = \frac{\sum_{i=1}^n \mu_i \cdot z_i}{\sum_{i=1}^n \mu_i}$$

Dimana y adalah nilai *crisp* dan μ_i adalah derajat keanggotaan dari nilai *crisp* y.

3. Perancangan Sistem

Dalam perancangan sistem prediksi dengan menggunakan metode *Hidden Markov Model* (HMM) dan *Fuzzy Logic* (FL) ini menggunakan diagram *flowchart* untuk menggambarkan pemodelan sistem yang dibangun.

Berikut gambaran perancangan sistem prediksi harga saham:



Gambar 3-1. Flowchart sistem prediksi harga saham

Berikut penjelasan mengenai flowchart perancangan sistem prediksi yang dibangun :

1. Data *training* yang diinputkan berupa nilai harga *open, high, low, close* dari data saham PT Telkom
2. Melakukan proses *training* HMM menggunakan data *training*.
3. Model HMM yang dihasilkan pada proses *training* akan digunakan untuk menghitung nilai *log-likelihood* dari data *testing*.
4. Data *testing* yang memiliki nilai *log-likelihood* yang telah dihasilkan akan dikelompokkan ke dalam *bucket-bucket*.
5. Data-data pada *bucket* akan digunakan untuk memprediksi harga *closing* saham hari berikutnya menggunakan *fuzzy* dengan bantuan *Gradient Descent* untuk mengoptimasi aturan *fuzzy*.

4. Analisis Hasil Pengujian

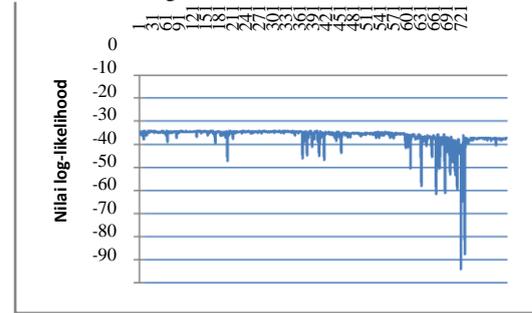
Proses pengujian sistem ini dilakukan dengan tujuan yang ingin dicapai sebagai berikut:

1. Menganalisis performansi HMM dalam mengelompokkan data
2. Menganalisis pengaruh perubahan jumlah *bucket* yang dibentuk dalam memprediksi harga saham.

Pada skenario pertama, yaitu performansi HMM dalam mengelompokkan data, akan diperlihatkan data-data yang ada di tiap *bucket*. Untuk skenario kedua, yaitu

pengaruh jumlah *bucket* terhadap sistem, jumlah *bucket* yang akan dibentuk yaitu 2, 4, dan 8. Sebagai *default*, jumlah *bucket* = 8, jumlah *state* = 4, *threshold* = 0.0001, dan maksimum iterasi = 30. Data saham yang digunakan pada kedua skenario diatas terbagi atas 3 tahun data *training* dan 1 tahun data *testing*.

1. Performansi HMM dalam mengelompokkan data



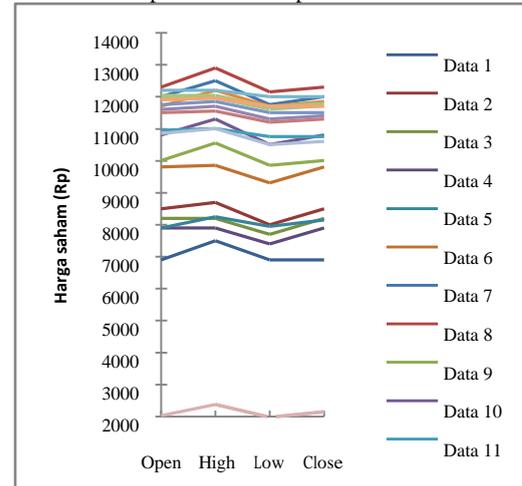
Gambar 4-1 Nilai log-likelihood data training

Setelah dikelompokkan, *bucket* yang terbentuk adalah seperti berikut:

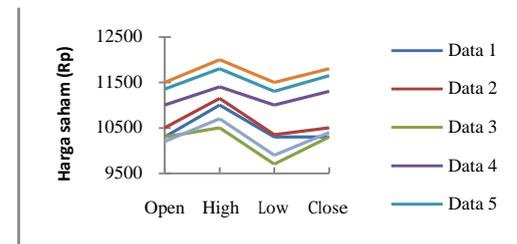
Tabel 4-1 Jumlah data *training* pada tiap *bucket*

Bucket ke-	1	2	3	4	5	6	7	8
Jumlah data	2	0	0	1	4	7	20	696

Berikut adalah pola untuk data pada *bucket* 7 dan 6:



Gambar 4-2 Pola data pada *bucket* 7 untuk data training



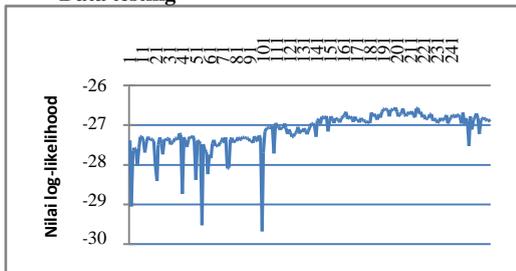
Gambar 4-3 Pola data pada *bucket* 6 untuk data training

Berdasarkan Tabel 4-1, persebaran data pada *bucket* tidak merata. Hal ini dapat dilihat pada jumlah data di *bucket* terakhir, yang menampung data paling banyak. *Bucket* kedelapan memiliki 696 data, yaitu 95.34% dari data keseluruhan. *Bucket* ketujuh dan keenam memiliki data yang paling banyak secara berurutan setelah *bucket* pertama. *Bucket* ketujuh memiliki 20 data, 2.73% dari

data keseluruhan. *Bucket* keenam memiliki 7 data, 0.95% dari data keseluruhan. Sisa data terbagi ke *bucket* 1, 4, dan 5. Sedangkan *bucket* 2 dan 3 tidak memiliki data.

Ketidakrataan persebaran data dikarenakan oleh *range* nilai *log-likelihood* yang besar, mulai dari -24.33 sampai dengan -84.26. *Range* nilai *log-likelihood* yang besar ini disebabkan oleh adanya beberapa data yang memiliki nilai *log-likelihood* yang berdekatan juga merupakan salah satu faktor dalam ketidakrataan persebaran data pada *bucket*. Hal ini dikarenakan kurang mampunya model HMM yang telah dibangun pada proses training dalam mengelompokkan data training. Jika kita memperhatikan Gambar 4-2, akan terlihat bahwa kebanyakan data yang berada di *bucket* 7 memiliki pola yang sama. Begitu juga dengan data yang terlihat di Gambar 4-3. Namun, jika diperhatikan dengan seksama, pola data pada *bucket* 6 dan *bucket* 7 mirip. Ini menandakan bahwa data-data di dua *bucket* tersebut memiliki pola yang sama. Dapat disimpulkan bahwa banyak data-data pada data *training* memiliki kemiripan pola, dan pola di data *training* tidak beragam.

• **Data testing**



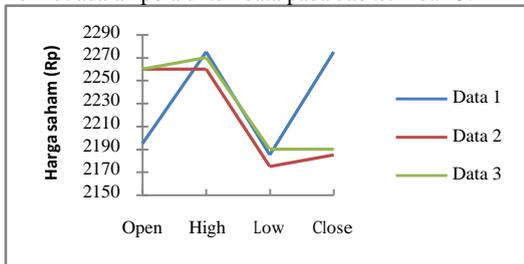
Gambar 4-4 Nilai log-likelihood data testing

Setelah dikelompokkan, *bucket* yang terbentuk adalah seperti berikut:

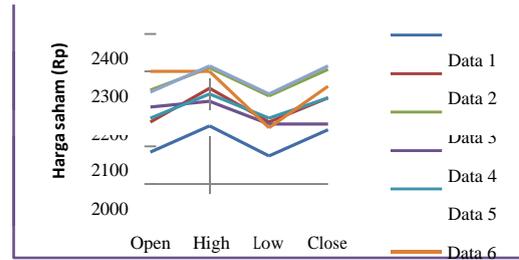
Tabel 4-2 Jumlah data testing pada tiap bucket

Bucket ke-	1	2	3	4	5	6	7	8
Jumlah data	2	1	1	3	7	55	67	105

Berikut adalah pola untuk data pada *bucket* 4 dan 5:



Gambar 4-5 Pola data pada bucket 4 untuk data testing



Gambar 4-6 Pola data pada bucket 5 untuk data testing

Seperti yang dapat dilihat di Tabel 4-2, jumlah data di *bucket* 6, 7, dan 8 jauh lebih banyak daripada jumlah data di *bucket* lainnya. *Bucket* kedelapan memiliki 105 data, yaitu 43.56% dari total data. Persentase jumlah data pada *bucket* ketujuh dan keenam untuk data *testing* tidak sesedikit pada data *training*. Pada *bucket* ketujuh, terdapat 67 data, 27.8%. Pada *bucket* keenam, terdapat 55 data, 22.82%. Sisa data terbagi kepada sisa *bucket*. Tidak seperti data *training*, *bucket* pada data *testing* terisi semua. Hal ini menandakan bahwa data tersebar lebih merata daripada data *training*. Namun *bucket* pertama tetap memiliki jumlah data terbanyak.

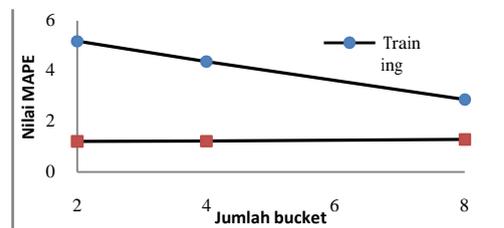
Tidak seperti data *training* yang memiliki *range* nilai *log-likelihood* yang besar, *range* nilai *log-likelihood* data *testing* cukup kecil, mulai dari -26.55 sampai dengan -29.68. Hal ini yang mengakibatkan persebaran data yang lebih rata daripada data *training*. Model HMM yang telah dibangun lebih bagus dalam mengelompokkan data *testing* daripada data *training*.

Jika kita memperhatikan Gambar 4-5, akan terlihat bahwa hampir semua data yang berada di *bucket* 5 memiliki pola yang sama. Begitu juga dengan data yang terlihat di Gambar 4-6. Namun, jika diperhatikan dengan seksama, pola data pada *bucket* 4 dan *bucket* 5 mirip. Ini menandakan bahwa data-data di dua *bucket* tersebut memiliki pola yang sama. Dapat disimpulkan bahwa banyak data-data pada data *testing* memiliki kemiripan pola, dan pola di data *testing* tidak beragam.

2. Pengaruh jumlah bucket yang dibentuk terhadap performansi sistem

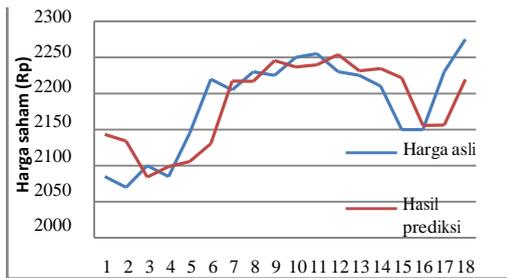
Tabel 2-1 MAPE (%) terbaik untuk skenario pengaruh jumlah bucket

Jumlah bucket	2	4	8
Training	5.189564	4.374294	2.867821
Testing	1.212999	1.234109	1.290048



Gambar 2-1 Nilai MAPE terbaik untuk jumlah bucket berbeda

Berikut adalah perbandingan harga asli dan hasil prediksi pada pengujian dengan 2 *bucket*:



Gambar 2-2 Perbandingan harga asli dan hasil prediksi saham Telkom Januari 2014

Berdasarkan Tabel 4-3, terlihat bahwa nilai MAPE untuk *testing set* lebih bagus daripada nilai MAPE untuk *training set* untuk semua kasus. Hal ini menunjukkan bahwa sistem yang dibentuk mampu menangani data *testing* lebih baik daripada data *training*. Dapat dilihat juga bahwa untuk data *training*, nilai MAPE semakin kecil dengan banyaknya jumlah *bucket*. Sedangkan untuk data *testing*, semakin banyak jumlah *bucket*, semakin tinggi nilai MAPE-nya. Walaupun begitu, perubahan yang terjadi tidak terlalu signifikan seperti yang terjadi di data *training*.

5. Kesimpulan dan Saran

6.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan, antara lain :

1. Penggunaan HMM dan *Fuzzy Model* dapat digunakan untuk memprediksi harga saham.
2. Model HMM yang dibangun pada proses training kurang dapat mengelompokkan data saham Telkom. Hal ini dapat dilihat dengan persebaran data yang tidak merata di *bucket* dan terfokus pada *bucket* terakhir. Untuk data training, *bucket* kedelapan memiliki 696 data, yaitu 95.34% dari data keseluruhan. Untuk data testing, *bucket* kedelapan memiliki 105 data, yaitu 43.56%
3. Dalam kasus ini, untuk data training, semakin banyak jumlah *bucket* yang dibentuk, semakin berkurang nilai MAPE-nya. Nilai MAPE pada pengujian dengan 2 *bucket* adalah 5.19, pada pengujian dengan 4 *bucket* adalah 4.37, dan pada pengujian dengan 4 *bucket* adalah 2.87.
4. Dalam kasus ini, untuk data testing, semakin banyak jumlah *bucket* yang dibentuk, semakin bertambah nilai MAPE-nya. Namun penambahan yang terjadi tidaklah signifikan. Nilai MAPE pada pengujian dengan 2 *bucket* adalah 1.21, pada pengujian dengan 4 *bucket* adalah 1.23, dan pada pengujian dengan 4 *bucket* adalah 1.29.
5. Sistem dapat menangani data testing lebih baik daripada data training. Nilai MAPE terbaik untuk data testing adalah 1.21, sedangkan nilai MAPE terbaik untuk data training adalah 2.87.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan selanjutnya:

1. Menggunakan data saham yang memiliki lebih banyak pola sehingga persebaran data pada proses pengelompokkan ke *bucket* dapat lebih rata.

Daftar Pustaka

- [1] Agusta, Yudi, 2008. *Hidden Markov Model* <http://yudiagusta.wordpress.com/2008/06/25/hidden-markov-models/>
- [2] Firdaniza, Nurul Gusriani dan Akmal. 2006. *Hidden Markov Model*. Jatinangor: Universitas Padjajaran.
- [3] Furi, Ratih Puspita. 2015. *Prediksi Financial Time Series Menggunakan Independent Component Analysis dan Support Vector Regression. Studi Kasus : IHSG dan JII*. Bandung: Universitas Telkom.
- [4] Hasan, Muhammad Amiruddin. 2014. *Prediksi Harga Saham Menggunakan Kombinasi Wavelet Transform dan Artificial Neural Network*. Bandung: Universitas Telkom.
- [5] Hassan, Md. Rafiul and Nath, Baikunth. 2005. *Stock Market Forecasting Using Hidden Markov Model: A New Approach*. Australia: The University of Melbourne.
- [6] Hassan, Md. Rafiul. 2009. *A combination of hidden Markov model and fuzzy model for stock market forecasting*. Australia: The University of Melbourne.
- [7] Purnomo, Danu Sogi. 2013. *Analisis dan Implementasi Elman Recurrent Neural Network dan Firefly Algorithm pada Prediksi Harga Minyak Mentah*. Bandung: Universitas Telkom.
- [8] Rabiner, L.R.. 1989. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE 77 257-286.
- [9] Sasongko, Alfian Yudha. 2015. *Analisis dan Implementasi LS-SVM (Least Square Support Vector Machine) dengan Studi Kasus Kisaran Harga Saham*. Bandung: Universitas Telkom.
- [10] Suyanto, 2007, *Artificial Intelligence Searching, Reasoning, Planning and Learning*. Bandung: Informatika Bandung.
- [11] Zhang, Yingjian. 2004. *Prediction of Financial Time Series With Hidden Markov Models*. Canada: Simon Fraser University.
- [12] Bursa Efek Indonesia. *Saham*. <http://www.idx.co.id/id-id/beranda/informasi/bagiinvestor/saham.aspx> (diakses 4 Juli 2015)
- [13] Elearning Universitas Gunadarma, *Perkembangan Pasar Modal* http://www.elearning.gunadarma.ac.id/docmodul/pe- ngenalan_pasar_modal/KPM.pdf (diakses tanggal 5 Februari 2015)
- [14] Elearning Universitas Widyagama Malang (2008). *Logika Fuzzy* http://k12008.widyagama.ac.id/ai/diktatpdf/Logika_Fuzzy.pdf (diakses tanggal 11 November 2012)