

# **IMPLEMENTASI ALGORITME RANDOM FOREST UNTUK PREDIKSI PERGERAKAN HARGA SAHAM**

## **TUGAS AKHIR**

Diajukan sebagai syarat untuk menyelesaikan  
Program Studi Strata-1 Departemen Informatika

**Disusun Oleh:**  
**Febrianda Rifqi Pratama**  
**1113055**



**INSTITUT  
TEKNOLOGI  
HARAPAN  
BANGSA**

*Veritas vos liberabit*

**DEPARTEMEN INFORMATIKA  
INSTITUT TEKNOLOGI HARAPAN BANGSA  
BANDUNG  
2017**



---

**LEMBAR PENGESAHAN**

**IMPLEMENTASI ALGORITME RANDOM FOREST UNTUK PREDIKSI  
PERGERAKAN HARGA SAHAM**



Disusun oleh:

Nama : Febrianda Rifqi Pratama

NIM : 1113055

Telah Disetujui dan Disahkan  
Sebagai laporan Tugas Akhir Departemen Informatika  
Institut Teknologi Harapan Bangsa

Bandung, Desember 2017

Disetujui,

Pembimbing

**Ir. Inge Martina, M.T.**

**NIK. 107013**



### PERNYATAAN HASIL KARYA PRIBADI

Saya yang bertanda tangan di bawah ini:

Nama : Febrianda Rifqi Pratama

NIM : 1113055

Dengan ini menyatakan bahwa laporan Tugas Akhir dengan Judul : ”**IMPLEMENTASI ALGORITME RANDOM FOREST UNTUK PREDIKSI PERGERAKAN HARGA SAHAM**” adalah hasil pekerjaan saya dan seluruh ide, pendapat atau materi dari sumber lain telah dikutip dengan cara penulisan referensi yang sesuai.

Pernyataan ini saya buat dengan sebenar-benarnya dan jika pernyataan ini tidak sesuai dengan kenyataan maka saya bersedia menanggung sanksi yang akan dikenakan pada saya.

Bandung, Desember 2017

Yang membuat pernyataan,

Febrianda Rifqi Pratama

## **PEDOMAN PENGGUNAAN TUGAS AKHIR**

Laporan tugas akhir yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Institut Teknologi Harapan Bangsa, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada pengarang dan pembimbing Tugas Akhir. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan dengan seizin pengarang atau pembimbing Tugas Akhir dan harus disertai dengan ketentuan penulisan ilmiah untuk menyebutkan sumbernya.

Tidak diperkenankan untuk memperbanyak atau menerbitkan sebagian atau seluruh laporan tugas akhir tanpa seizin dari pengarang atau pembimbing Tugas Akhir yang bersangkutan.

## KATA PENGANTAR

Terima kasih kepada Tuhan yang Maha Esa karena dengan bimbingan-Nya dan karunia-Nya penulis dapat melaksanakan Tugas Akhir yang berjudul "IMPLEMENTASI ALGORITME RANDOM FOREST UNTUK PREDIKSI PERGERAKAN HARGA SAHAM". Laporan ini disusun sebagai salah satu syarat kelulusan di Institut Teknologi Harapan Bangsa. Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, karena oleh bimbingan-Nya penulis selalu mendapat pengharapan untuk menyelesaikan tugas akhir ini.
2. Ibu Ir. Inge Martina, M.T., selaku pembimbing I Tugas Akhir yang senantiasa memberi dukungan, semangat, ilmu-ilmu, saran dan dukungan kepada penulis selama tugas akhir berlangsung dan selama pembuatan laporan tugas akhir ini.
3. Ibu Ria Chaniago, S.T, M.T., Ibu Elisafina Siswanto, S.T., M.T., selaku penguji I Tugas Akhir. Terima kasih atas dukungan, semangat, ilmu-ilmu, dan masukan yang telah diberikan kepada penulis dalam menyelesaikan Laporan Tugas Akhir ini
4. Ibu Mentiana Sibarani SE., M.Si, selaku penguji II Tugas Akhir. Terima kasih atas dukungan, semangat, ilmu-ilmu, dan masukan yang telah diberikan kepada penulis dalam menyelesaikan Laporan Tugas Akhir ini
5. Seluruh dosen dan staff Departemen Informatika ITHB yang telah membantu dalam menyelesaikan Laporan Tugas Akhir ini.
6. Segenap jajaran staf dan karyawan ITHB yang turut membantu kelancaran dalam menyelesaikan Laporan Tugas Akhir ini.
7. Kedua orang tua tercinta yang selalu menyediakan waktu untuk memberikan doa, semangat dan dukungan yang tak habis-habisnya kepada penulis untuk menyelesaikan Laporan Tugas Akhir ini. Terima kasih untuk nasihat, masukan, perhatian, teguran dan kasih sayang yang diberikan hingga saat ini.

8. Kepada Ana, Alvin, Aghnat, Deniar, Fazari, Hilda, Khadir, Rizka, Salma, dan Tika.

Terimakasih kalian selalu memberikan semangat, doa, dan dukungan kepada penulis selama ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna karena keterbatasan waktu dan pengetahuan yang dimiliki oleh penulis. Oleh karena itu, kritik dan saran untuk membangun kesempurnaan tugas akhir ini sangat diharapkan. Semoga tugas akhir ini dapat membantu pihak-pihak yang membutuhkannya.

Bandung, Desember 2017

Hormat Saya,

Febrianda Rifqi Pratama.

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b>	<b>i</b>
<b>LEMBAR PERNYATAAN HASIL KARYA PRIBADI</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>PEDOMAN PENGGUNAAN TUGAS AKHIR</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>iv</b>
<b>DAFTAR ISI</b>	<b>viii</b>
<b>DAFTAR TABEL</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>x</b>
<b>I PENDAHULUAN</b>	<b>1-1</b>
1.1 Latar Belakang . . . . .	1-1
1.2 Rumusan Masalah . . . . .	1-2
1.3 Tujuan Penelitian . . . . .	1-2
1.4 Batasan Masalah . . . . .	1-2
1.5 Kontribusi Penelitian . . . . .	1-2
1.6 Metode Penelitian . . . . .	1-2
1.7 Sistematika Penulisan . . . . .	1-3
<b>II LANDASAN TEORI</b>	<b>2-1</b>
2.1 <i>Forecasting</i> . . . . .	2-1
2.2 <i>Machine Learning</i> . . . . .	2-2
2.3 <i>Supervised Learning</i> . . . . .	2-3
2.4 Pemilihan Fitur . . . . .	2-3

2.4.1	Diskritasi . . . . .	2-5
2.5	<i>Random Forest</i> . . . . .	2-5
2.5.1	<i>Decision Tree</i> . . . . .	2-6
2.6	<i>F-Measure</i> . . . . .	2-6
2.7	TA4J . . . . .	2-7
2.8	Tinjauan Pustaka . . . . .	2-10
2.8.1	Saham . . . . .	2-10
2.8.2	Analisis Saham . . . . .	2-11
2.8.3	Indikator Teknis Saham . . . . .	2-13
2.8.3.1	<i>Moving Average</i> . . . . .	2-13
2.8.3.2	<i>Stochastic Oscilliator</i> . . . . .	2-14
2.8.3.3	<i>Williams' Percent Range</i> . . . . .	2-15
2.8.3.4	<i>Moving Average Convergence Divergence</i> . . . . .	2-16
2.8.3.5	<i>Accumulation/Distribution Oscillator</i> . . . . .	2-17
2.8.3.6	<i>Relative Strength Index</i> . . . . .	2-18
2.8.3.7	<i>Community Channel Index</i> . . . . .	2-19
2.8.3.8	<i>Momentum</i> . . . . .	2-21
2.8.3.9	<i>Alligator Indicator</i> . . . . .	2-22

<b>III ANALISIS DAN PERANCANGAN SISTEM</b>	<b>3-1</b>	
3.1	Analisis Masalah . . . . .	3-1
3.2	Kerangka Pemikiran . . . . .	3-2
3.3	Tahap Analisis Urutan Proses Umum . . . . .	3-3
3.3.1	Analisis Data . . . . .	3-4
3.3.2	Ekstraksi Fitur . . . . .	3-4
3.3.3	Proses <i>Learning</i> . . . . .	3-4
3.3.4	Klasifikasi . . . . .	3-5
3.4	Perancangan Kasus . . . . .	3-6
3.4.1	<i>Feature Extraction</i> . . . . .	3-6
3.4.2	<i>Proses Learning</i> . . . . .	3-12
3.4.3	Klasifikasi . . . . .	3-17

<b>IV IMPLEMENTASI DAN PENGUJIAN</b>	<b>4-1</b>
4.1 Lingkungan Implementasi . . . . .	4-1
4.1.1 Spesifikasi Perangkat Keras . . . . .	4-1
4.1.2 Lingkungan Perangkat Lunak . . . . .	4-1
4.2 Implemenatai Perangkat Lunak . . . . .	4-1
4.2.1 Class Diagram . . . . .	4-2
4.2.2 Daftar <i>Method</i> . . . . .	4-2
4.2.3 Implementasi Metode . . . . .	4-5
4.2.3.1 Inisialisasi Data . . . . .	4-5
4.2.3.2 Implementasi Ekstraksi Fitur . . . . .	4-5
4.2.3.3 Implementasi Proses <i>Learning</i> . . . . .	4-5
4.2.3.4 Implementasi Klasifikasi . . . . .	4-6
4.3 Pengujian . . . . .	4-6
4.3.1 Ketentuan Pengujian . . . . .	4-6
4.3.2 Pengujian <i>Random Forest</i> . . . . .	4-7
4.3.2.1 Pengujian Jumlah Pohon Keputusan . . . . .	4-7
4.3.2.2 Pengujian Jumlah Fitur Data . . . . .	4-10
<b>V PENUTUP</b>	<b>5-1</b>
5.1 Kesimpulan . . . . .	5-1
5.1.1 Saran . . . . .	5-1
<b>DAFTAR PUSTAKA</b>	<b>xii</b>

## **DAFTAR TABEL**

2.1	Fungsi pada TA4J yang digunakan . . . . .	2-7
2.1	Fungsi pada TA4J yang digunakan . . . . .	2-8
2.1	Fungsi pada TA4J yang digunakan . . . . .	2-9
3.1	Contoh Data Historis Saham . . . . .	3-6
3.2	Contoh hasil ekstraksi fitur data . . . . .	3-9
3.3	Contoh hasil diskritasi fitur data . . . . .	3-11
3.4	Contoh Data Training . . . . .	3-12
3.5	Contoh Data Testing . . . . .	3-17
3.6	Contoh Data Testing . . . . .	3-18
4.1	Daftar Method . . . . .	4-2
4.1	Daftar Method . . . . .	4-3
4.1	Daftar Method . . . . .	4-4
4.2	Nilai Parameter Indikator Teknis Yang Digunakan . . . . .	4-7
4.3	Pengujian Jumlah Pohon Keputusan implementasi tipe A . . . . .	4-8
4.4	Pengujian Jumlah Pohon Keputusan implementasi tipe B . . . . .	4-8
4.4	Pengujian Jumlah Pohon Keputusan implementasi tipe B . . . . .	4-9
4.5	Pengujian Jumlah Fitur Data implementasi tipe A . . . . .	4-10
4.6	Pengujian Jumlah Fitur Data implementasi tipe B . . . . .	4-10
4.6	Pengujian Jumlah Fitur Data implementasi tipe B . . . . .	4-11
4.7	Pengujian Jumlah Fitur Data implementasi tipe A . . . . .	4-13
4.8	Pengujian Jumlah Fitur Data implementasi tipe B . . . . .	4-13
4.8	Pengujian Jumlah Fitur Data implementasi tipe B . . . . .	4-14

## DAFTAR GAMBAR

2.1	Contoh Grafik Indikator <i>Moving Average</i> [3] . . . . .	2-14
2.2	Contoh Grafik Indikator STC%K dan STC%D[3] . . . . .	2-15
2.3	Contoh Grafik Indikator <i>Williams' %R</i> [3] . . . . .	2-16
2.4	Contoh Grafik Indikator MACD[3] . . . . .	2-17
2.5	Contoh Grafik Indikator A/D[3] . . . . .	2-18
2.6	Contoh Grafik Indikator RSI[3] . . . . .	2-19
2.7	Contoh Grafik Indikator CCI[3] . . . . .	2-21
2.8	Contoh Grafik Indikator <i>Momentum</i> [3] . . . . .	2-22
2.9	Contoh Grafik Indikator <i>Alligator</i> [3] . . . . .	2-23
3.1	Diagram Kerangka Pemikiran . . . . .	3-2
3.2	Diagram Proses Umum . . . . .	3-3
3.3	Diagram Proses Learning . . . . .	3-5
3.4	Contoh Hasil Proses <i>Learning</i> Pohon Keputusan . . . . .	3-17
4.1	Class Diagram . . . . .	4-2
4.2	Hasil Analisis Fitur Dataset UNVR Tanpa Pemilihan Fitur Data . . . . .	4-11
4.3	Hasil Analisis Fitur Dataset UNVR Implementasi Tipe A . . . . .	4-11
4.4	Hasil Analisis Fitur Dataset UNVR Implementasi Tipe B . . . . .	4-12
4.5	Hasil Analisis Fitur Dataset GGRM Tanpa Pemilihan Fitur Data . . . . .	4-12
4.6	Hasil Analisis Fitur Dataset GGRM Implementasi Tipe A . . . . .	4-12
4.7	Hasil Analisis Fitur Dataset GGRM Implementasi Tipe B . . . . .	4-12

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Banyak investor yang menginvestasikan uangnya dalam jumlah yang besar, namun resiko dalam berinvestasi pada pasar saham sangat besar dikarenakan pergerakan pasar saham yang selalu berubah-ubah dan tidak pernah berhenti. Dibutuhkan analisa secara mendalam untuk dapat memprediksi pergerakan pasar saham. Dua jenis analisa yang dilakukan investor untuk menentukan pergerakan harga saham adalah analisa fundamental dan analisa teknis. Para investor sangat bergantung kepada hasil analisa indikator-indikator saham seperti *Moving Average*(MA), *Moving Average Convergence/Divergence*(MACD), *Relative Strength Index*, dan sebagainya[1]. Masalah utama dari analisa indikator saham tersebut adalah setiap indikator teknis menganalisa dan menginterpretasikan harga dengan cara yang berbeda[2]. Setiap teknik analisa teknis adalah hasil dari pengamatan banyak analis, dan para analis mengamati pergerakan saham dari perspektif masing-masing. Ada faktor subjektifitas yang dapat mempengaruhi kepastian dari hasil akhir analisis.

Para analis akan memprediksi pergerakan harga saham dengan tidak melawan pergerakan tren saham[3]. Setiap perubahan tren saham akan menuntun naik dan turunnya harga saham di pasar. Analisa cukup dilakukan dengan menggunakan teknik analisa teknis karena harga saham dapat bereaksi cepat dengan kejadian di pasar saham[4]. Hal tersebut dapat mencegah investor dalam mengalami kerugian yang besar dan dapat digunakan oleh perusahaan untuk mengetahui tren pasar pada masa yang akan datang. Salah satu kesulitan dalam mengolah informasi dari analisa teknis adalah kekhawatiran hilangnya nilai atau makna dari informasi yang dibawa oleh indikator-indikator teknis ketika digunakan sebagai input model prediksi[5]. Model prediksi harus mengorelasikan antara trend masukkan dengan tren keluaran. Untuk mengatasi hal tersebut, maka nilai-nilai indikator analisis teknis akan diubah ke dalam bentuk diskrit(1, -1) sesuai dengan perilaku setiap indikator teknis

Untuk mengolah informasi dari data historis saham dibutuhkan model prediksi *non-linear* seperti ANN, SVM, dan Random *Forest*. Metode ANN[6][7][8] bekerja dengan cara membuat jaringan seperti jarikan otak manusia. ANN memiliki kelebihan dalam beradaptasi terhadap pola data dan hubungan antara masukkan dan hasil prediksi data. Karena kemampuan adaptasinya, ANN digunakan dalam berbagai bidang. SVM adalah salah satu metode yang menjadi alternatif dari ANN. SVM[9][10][11] melakukan klasifikasi dengan membangun ruang dimensi fitur yang besar. SVM menggunakan fungsi kernel untuk menangani masalah non-linear secara efisien. SVM dapat memberikan akurasi yang lebih baik daripada ANN[5].

*Random Forest*[3] adalah salah satu metode *ensamble learning* dimana banyak pohon keputusan sebagai *weak learner* akan dibuat dan dikumpulkan menjadi sebuah kumpulan besar dan setiap hasil keputusannya akan diperoleh melalui hasil rata-rata kumpulan *weak learner* tersebut. Salah satu keuntungan dari *Random Forest* adalah proses *bagging* yang dapat mengurangi terjadinya *over-fitting* data. Penelitian menunjukkan bahwa *Random Forest* mampu memberikan hasil lebih baik dari ANN dan SVM[3].

### 1.2 Rumusan Masalah

Masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana pengaruh setiap indikator terhadap hasil uji?
2. Bagaimana pengaruh jumlah pohon keputusan yang dibuat terhadap hasil uji?

### 1.3 Tujuan Penelitian

Membuat sebuah aplikasi yang dapat memprediksi trend saham dengan menggunakan metode *Random Forest*.

### 1.4 Batasan Masalah

Penelitian tidak membahas teknik analisis fundamental.

### 1.5 Kontribusi Penelitian

Pada penelitian sebelumnya[3], *Random Forest* dapat memberikan hasil lebih baik dari ANN dan SVM untuk memprediksi tren saham. Oleh karena itu, penulis menerapkan metode prediksi *Random Forest* kepada indeks saham perusahaan di Indonesia dan menganalisa parameter optimal bagi metode prediksi. Penulis juga menambahkan indikator *alligator* untuk mengidentifikasi tren saham yang sedang terjadi.

### 1.6 Metode Penelitian

Penelitian ini akan dibagi menjadi lima tahap :

1. Studi Literatur

Pada tahap ini dilakukan proses pencarian bahan referensi melalui buku, jurnal/paper, artikel dan website-website terkait.

2. Analisis

Melakukan analisis data berdasarkan studi literatur yang sudah didapatkan. Dari tahap ini akan diperoleh kesimpulan untuk mengerjakan penelitian ini

3. Perancangan

Melibuti perancangan program dan algoritme yang digunakan untuk menyelesaikan permasalahan yang telah dianalisis.

4. Implementasi

Melakukan implementasi untuk menyelesaikan perancangan yang sudah ditetapkan berdasarkan metode yang telah dipilih oleh peneliti.

5. Pengujian

Melakukan pengujian terhadap metode yang digunakan untuk mengetahui apakah metode tersebut yang layak dan sesuai.

### **1.7 Sistematika Penulisan**

Pada penelitian ini peneliti menyusun berdasarkan sistematika penulisan sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan metodologi penelitian.

#### **BAB II LANDASAN TEORI**

Bab ini berisi tentang teori-teori yang mendukung atau berhubungan dengan penelitian ini.

#### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan tentang proses menganalisa masalah dan pemodelan dari penelitian ini.

#### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi tentang implementasi sistem dan pengujian terhadap sistem yang telah dirancang.

#### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan ini dan saran untuk pengembangan lebih lanjut dari penelitian ini.

## BAB II

### LANDASAN TEORI

#### 2.1 *Forecasting*

*Forecasting* adalah sebuah kegiatan meramalkan masa depan seakurat mungkin, dengan menggunakan informasi yang tersedia, termasuk data historis dan pengetahuan tentang kejadian di masa depan yang akan mempengaruhi hal yang sedang diramalkan[12]. Kemungkinan meramalkan suatu kejadian atau kuantitas bergantung kepada beberapa faktor, termasuk[12]:

1. Bagaimana kita memahami faktor yang mempengaruhinya.
2. Mengetahui berapa banyak data yang tersedia.
3. Memahami bagaimana prakiraan tersebut dapat mempengaruhi hal yang ingin diramalkan.

Dalam *forecasting*, hal paling utama adalah mengetahui kapan sesuatu dapat diramalkan secara akurat, dan kapan prakiraan hanya akan sama saja seperti melempar koin. Banyak orang keliru menganggap bahwa *forecasting* tidak akan mungkin terjadi dalam lingkungan yang selalu berubah. Setiap lingkungan akan selalu berubah, dan model peramalan yang baik dapat menangkap pola dari sebuah hal yang selalu berubah. Jarang sekali saat memprakirakan sesuatu menggunakan asumsi bahwa lingkungan tidak akan pernah berubah. Hal yang biasa diambil saat memprakirakan sesuatu adalah bagaimana sebuah lingkungan akan selalu berubah dan akan terus berubah di masa yang akan datang.

*Forecasting* selalu menjadi bagian utama dalam kegiatan pengambilan keputusan manajemen, karena hal itu dapat berperan penting pada berbagai bidang dalam sebuah perusahaan. Organisasi modern membutuhkan prakiraan jangka pendek, jangka menengah, atau jangka panjang, tergantung dari kebutuhannya. Pada awal tahap *Forecasting*, kita harus mengetahui apa yang akan diramalkan.

Beberapa hal yang dapat dipertimbangkan untuk mengetahui apakah *forecasting* dibutuhkan[12]:

1. Seluruh produk, atau hanya kelompok produk tertentu?
2. Seluruh gerai penjualan, atau berdasarkan gerai penjualan dari daerah tertentu, atau hanya total penjualan saja?
3. Data mingguan, bulanan, atau tahunan?

Diperlukan juga untuk mempertimbangkan periode *forecasting*. Apakah *forecasting* akan dibutuhkan untuk 1 bulan selanjutnya, untuk 6 bulan, atau untuk 10 tahun? Berdasarkan periode *forecast* akan sangat penting untuk mengetahui jenis model *forecast* yang akan cocok digunakan. Jika memerlukan hasil *forecasting* dalam rentang waktu yang singkat maka menggunakan sistem yang dapat menghasilkan hasil *forecast* secara otomatis akan lebih baik

dari pada menggunakan metode manual yang membutuhkan proses yang sangat berhati-hati.

Metode *forecasting* sangat bergantung kepada ketersediaan data yang ada. Pada umumnya, *quantitative prediction* menggunakan data *time-series* (diperoleh pada interval waktu tertentu), atau data *cross-sectional* (diperoleh pada 1 waktu saja). *Quantitative forecasting* dapat diterapkan jika dua kondisi ini terpenuhi:

1. Informasi numerik tentang masa lalu tersedia.
2. Dapat memungkinkan untuk mengasumsikan bahwa beberapa aspek dari pola yang terjadi di masa lampau akan terus terjadi di masa yang akan datang.

Jika tidak ada data yang tersedia atau tidak dapat di prakirakan, maka harus menggunakan metode *qualitative forecasting*.

*Forecasting* biasanya akan melakukan 5 tahap dasar[12]:

1. Mendefinisikan masalah

Mendefinisikan masalah membutuhkan pemahaman dari cara *forecasting* digunakan, siapa yang membutuhkan *forecasting*, dan bagaimana *forecasting* cocok dengan organisasi yang membutuhkan *forecasting*.

2. Mengumpulkan informasi

Ada sekurangnya 2 jenis informasi yang dibutuhkan: data statistik, dan pengetahuan para pakar yang mengumpulkan data dan menggunakan *forecast*. Biasanya akan sangat sulit untuk mengumpulkan cukup data historis yang cocok digunakan pada model statistik.

3. Analisis Awal

Selalu mulai dari menganalisis grafik dari data. Apakah memiliki pola yang konsisten? Apakah ada tren yang signifikan? Apakah ada tanda-tanda dari sebuah siklus bisnis? Apakah ada hal asing/ganjil pada data yang harus dianalisis oleh pakar ahli? Seberapa kuat hubungan antar variabel yang dianalisis?

4. Memilih dan mencocokan model.

Model yang paling cocok digunakan akan bergantung kepada ketersediaan data historis, hubungan antara variabel utama dan variabel pendukung, dan cara menggunakan *forecasts*.

5. Menggunakan dan mengevaluasi model *forecasting*

Performa model *forecast* akan dapat dievaluasi setelah data pada rentang waktu *forecast* sudah tersedia.

### 2.2 *Machine Learning*

*Machine learning* adalah sebuah program komputer yang mengoptimalkan kriteria kinerja dengan menggunakan contoh data atau pengalaman di masa lalu[13]. Penerapan *machine learning* kepada *database* yang besar disebut *data mining*. Tetapi *machine learning* bukan hanya sekedar masalah *database*, *machine learning* juga termasuk sebagai bagian dari kecerdasan buatan. Untuk menjadi cerdas, sebuah sistem yang berada pada lingkungan yang selalu berubah-ubah harus mempunyai kemampuan untuk belajar. Jika sebuah sistem dapat

belajar dan beradaptasi terhadap berbagai perubahan, maka perancang sistem tidak perlu meramalkan dan memberikan berbagai solusi untuk semua situasi yang mungkin terjadi.

Machine learning menggunakan teori statistik untuk membuat sebuah model matematika, karena tugas utama dari *machine learning* adalah menarik kesimpulan dari sebuah sampel. Model *machine learning* didefinisikan oleh beberapa parameter dan proses *learning* adalah sebuah kegiatan mengoptimalkan parameter dari model menggunakan data *training* atau pengalamannya di masa lalu. Model *machine learning* dapat menjadi *predictive* untuk membuat sebuah prediksi yang akan terjadi di masa yang akan datang, atau *descriptive* untuk memperoleh pengetahuan dari data, atau keduanya.

### 2.3 Supervised Learning

*Supervised learning* berkaitan dengan semua proses *machine learning* yang mempelajari fungsi dari tipe masukkan ke tipe keluaran dengan menggunakan data yang terdiri dari contoh yang mempunyai masukkan dan hasil.[14] Contoh dari *supervised learning* adalah klasifikasi dan regresi.

*Supervised learning* pada *machine learning* dapat dimodelkan sebagai berikut[13]:

$$y = g(x|\theta) \quad (2 . 1)$$

Dimana:

$X$  = Input

$g(\cdot)$  = Model machine learning

$\theta$  = Parameter machine learning

$Y$  = Output

$Y$  pada regresi adalah sebuah angka, dan pada klasifikasi adalah kode kelas(contoh:0,1).  $g(\cdot)$  adalah fungsi regresi atau pada klasifikasi adalah fungsi diskriminan untuk memisahkan sampel ke berbagai kelas berbeda. Program *machine learning* mengoptimalkan parameter,  $\theta$ , untuk meminimalkan *error* menggunakan perkiraan yang sedekat mungkin dengan nilai yang tepat pada data *training*.

### 2.4 Pemilihan Fitur

*Pemilihan fitur* adalah sebuah studi tentang algoritme untuk mengurangi dimensi data agar dapat meningkatkan performa *machine learning*[14]. Tujuannya adalah menghilangkan fitur yang tidak relevan dan yang berlebihan. Menghilangkan fitur yang tidak relevan dapat mempengaruhi akurasi. Fitur yang berlebihan adalah jenis dari fitur yang tidak relevan.

Perbedaannya adalah fitur ini menyiratkan keberadaan fitur lainnya. Setiap fitur akan relevan dengan fitur yang lainnya, namun menghilangkan salah satu dari fitur tersebut tidak akan mempengaruhi performa belajar.

Struktur dari sistem pemilihan fitur terdiri dari empat komponen dasar: masukkan, pencarian, evaluasi, dan hasil. Hasil dari pemilihan fitur dapat menjadi daftar *ranking* fitur atau *subset* dari fitur. Pada konteks belajar, masukkan kepada sistem pemilihan fitur adalah data yang *supervised* (semua sampel dikaitkan dengan label kelas seperti pada *supervised learning*), *unsupervised* (tidak ada label kelas yang tersedia pada *unsupervised learning*), dan beberapa sampel mempunyai label kelas dan beberapa yang tidak mempunya label kelas seperti pada *semi-supervised learning*. Beberapa model pemilihan fitur seperti *wrapper*, *filter*, atau *embedded* dapat digunakan tergantung dari bagaimana sistem pemilihan fitur berkerja dengan sistem belajar yang digunakan.

Untuk mencari fitur yang relevan dapat menggunakan 2 cara: *feature ranking* dan *subset selection*. *Feature ranking* mengurutkan fitur berdasarkan sifat intrinsik dari data agar fitur yang diurutkan teratas dapat dipilih. *Subset selection* memilih beberapa fitur dari keseluruhan fitur. Pemilihan fitur dapat dilakukan beberapa cara: *forward selection* (Mulai dari *subset* kosong kemudian memilih dari kumpulan fitur yang lengkap), *backward elimination* (Mulai dari *subset* yang penuh dengan fitur kemudian mengeluarkannya), dan *random* (isi fitur yang berada di *subset* dapat ditentukan. Bila jumlah memenuhi suatu batas pengukuran, maka fitur akan dikurangi, jika tidak maka akan ditambah).

Tiga model klasik dari pemilihan fitur adalah *filter*, *wrapper*, dan *embedded*. Penelitian menunjukkan bahwa menggunakan pemilihan fitur model *embedded* dapat meningkatkan performa belajar. Model *filter* bergantung kepada pengukuran sifat intrinsik data. Kesesuaian informasi dan konsistensi data adalah dua contoh pengukuran sifat data. Model *wrapper* menggunakan algoritme belajar (contoh: klasifikasi atau *clustering*) dalam menentukan kualitas fitur.

Pengaruh dari pemilihan fitur dapat diperiksa berdasarkan evaluasi empiris. Dua pertanyaan yang pasti berhubungan dengan klasifikasi adalah: apakah menggunakan fitur yang dipilih dapat menghasilkan hasil yang serupa seperti menggunakan seluruh fitur, dan bagaimana cara membandingkan dua algoritme pemilihan fitur untuk mencari mana yang lebih efektif. Salah satu cara yang biasa digunakan adalah dengan membandingkan akurasi klasifikasi. Jika tidak perbedaan yang signifikan maka tidak algoritme yang lebih baik dari keduanya. Jika ada perbedaan yang cukup signifikan, maka algoritme yang menghasilkan akurasi paling bagus adalah algoritme yang lebih efisien.

Masalah lain yang sering muncul dari evaluasi pemilihan fitur adalah bias dari pemilihan fitur. Dengan menggunakan data *training* yang sama pada pemilihan fitur dan proses belajar klasifikasi dapat menghasilkan bias seleksi. Menurut teori statistik berdasarkan penelitian regresi, bias dapat memperparah data *overfitting* dan berdampak negatif kepada performa klasifikasi. Disarankan untuk menggunakan kumpulan data yang berbeda untuk pemilihan fitur dan proses belajar. Namun, pada kenyataannya hal ini jarang dilakukan karena

ada keinginan untuk menggunakan data sebanyak mungkin untuk proses pemilihan fitur dan proses belajar.

### 2.4.1 Diskritasi

Diskritasi adalah sebuah proses dimana mengubah suatu nilai atribut data yang berbentuk numerik menjadi bentuk diskrit[14]. Dengan menggunakan proses diskritasi pada data dapat meningkatkan tingkat akurasi dan kecepatan komputasi pada model prediksi.

Diskritasi dapat digunakan juga untuk menjaga informasi yang dibawa oleh setiap indikator teknis tidak hilang ketika digunakan sebagai masukkan klasifikasi oleh model prediksi[5]. Teknik ini mengubah nilai setiap indikator teknis dari bentuk kontinu menjadi bentuk +1 dan -1. Dimana +1 menunjukkan tren saham sedang naik dan -1 menunjukkan tren saham sedang turun.

## 2.5 *Random Forest*

*Random Forest* adalah salah satu model *machine learning* yang membangun kumpulan besar dari pohon keputusan. *Random Forest* adalah modifikasi dari *bagging(bootstrap aggregating)* yang sebelumnya diciptakan oleh Breiman[15]. Menurut Breiman, *bagging* mampu menjadi model *machine learning* yang baik namun tidak seakurat yang ia harapkan.

Konsep dasar dari bagging adalah membangun kumpulan pohon keputusan dengan menggunakan sampel data yang berbeda. Setiap sampel data dibuat dengan cara memilih secara acak dari data *training* yang sebenarnya dan diperbolehkan adanya data yang sama pada pemilihan acak sampel data. Walaupun menggunakan sampel data yang berbeda, namun sampel data tidak bersifat independen karena berasal dari set data yang sama. Pemilihan acak subset data dilakukan agar meningkatkan tingkat perbedaan pada setiap pohon keputusan yang dibuat dan dengan meningkatkan perbedaan pada setiap pohon keputusan dapat akurasi model prediksi secara keseluruhan. Breiman menganalisa kembali bagging dan mendapatkan kesimpulan bahwa pohon keputusan yang dibuat dengan menggunakan *bagging* masih terlalu mirip satu sama lain dan hal tersebut dapat mengurangi akurasi model secara keseluruhan.

*Random Forest* menggunakan konsep seperti *bagging* namun pada *Random Forest* menambahkan proses pemilihan acak subset fitur pada proses *splitting* di setiap *node*[16]. Hal tersebut adalah ide Breiman untuk meningkatkan tingkat perbedaan setiap pohon keputusan dari model *bagging* dengan mengurangi korelasi antar pohon keputusan tanpa meningkatkan variansi dari model keseluruhan. Proses *pruning* pada setiap pohon keputusan di dalam *Random Forest* tidak perlu dilakukan karena jika pohon keputusan dibuat cukup dalam akan memiliki bias yang rendah, dan setiap pohon keputusan yang dibuat sudah terdistribusi secara identik, yang berarti bias dari keseluruhan pohon keputusan sama seperti bias dari salah satu dari pohon keputusan yang dibuat. Salah satu cara untuk meningkatkan akurasi adalah dengan mengurangi variansi dari seluruh pohon keputusan yang dibuat.

### 2.5.1 Decision Tree

Pohon keputusan adalah salah satu model klasifikasi yang membentuk struktur pohon, yang sangat mudah untuk dipahami, bahkan untuk pengguna yang awam, dan dapat menarik kesimpulan secara efisien dari data[17].

Proses belajar pohon keputusan dilakukan secara rekursif dan menggunakan metode divide-and-conquer. Algoritme pohon keputusan akan dimulai dari akar pohon, memilih atribut terbaik untuk memisahkan data menjadi beberapa set data berdasarkan nilai atribut yang dipilih pada setiap data, dan menambahkan cabang pada pohon keputusan. Pemilihan atribut terbaik akan dilakukan pada setiap cabang pohon hingga mencapai ujung dari cabang pohon keputusan.

Untuk memilih atribut terbaik untuk memisahkan data pada setiap cabang pohon, dapat menggunakan fungsi *impurity*. Fungsi *impurity* digunakan untuk mengukur tingkat kemungkinan sebuah cabang akan memiliki kumpulan data dengan kelas yang sama. Pengukuran *impurity* dapat dilakukan dengan menggunakan *entropy*. Dengan menghitung *impurity*, maka akan dapat diketahui seberapa besar sebuah atribut dapat memisahkan data dari *subset*. Penghitungan dilakukan dengan cara mencari atribut yang mampu mengurangi rata-rata *impurity* data *subset* paling besar, atau disebut sebagai *gain*.

Untuk menghitung gain, dapat menggunakan rumus berikut[17]:

$$\text{Entropy}(S) = - \sum_{n=1}^n \left( \frac{|S_i|}{|S|} \right) \log_2 \left( \frac{|S_i|}{|S|} \right) \quad (2 . 2)$$

$$\text{Gain}(S, A) = \text{Impurity}(S) - \sum_{t \in \text{Values}(A)} \frac{|S_i|}{|S|} \text{Impurity}(S_i) \quad (2 . 3)$$

Dimana:

$S$  = Data yang nilai *entropy* nya sedang dihitung.

$S_i$  = Total data dengan elemen dari label  $i$ .

$A$  = Atribut yang dipilih

## 2.6 F-Measure

*F-measure* adalah salah satu metode untuk mengukur tingkat keakuratan informasi yang dihasilkan oleh model prediksi[14]. Pengukuran dibagi menjadi 2 bagian. *Precision* mengukur jumlah data yang tepat diprediksi dibandingkan dengan jumlah data yang diprediksi dengan label yang sama. *Recall* mengukur jumlah data yang tepat diprediksi dibandingkan jumlah data yang memiliki label yang sama pada data *testing*.

*F-measure* dapat didefinisikan sebagai berikut:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2 . 4)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2 . 5)$$

$$F - Measure = \frac{2 * Recall * Precision}{(Recall + Precision)} \quad (2 . 6)$$

## 2.7 TA4J

*Library* TA4J(<https://github.com/ta4j/ta4j>) adalah kumpulan sintaks program yang dapat digunakan untuk menghitung berbagai rumus indikator teknis. Untuk menggunakan fitur *library* TA4J, sistem harus menginisialisasikan *object timeseries* yaitu sebuah *object* berisi kumpulan data historis saham yang sudah diformat sesuai dengan kebutuhan *library* TA4J. Kemudian *object timeseries* digunakan untuk membuat berbagai macam *object* indikator teknis yang berisi kumpulan data-data indikator teknis yang sudah dikalkulasi secara otomatis oleh *library* TA4J. Parameter periode perhitungan indikator teknis dapat diatur ketika membuat objek indikator teknis. *Object timeseries* dapat digunakan juga untuk membuat kumpulan data harga yang spesifik seperti kumpulan harga pembukaan, harga penutupan, dan sebagainya.

**Tabel 2.1** Fungsi pada TA4J yang digunakan

No	Nama Method	Input	Output	Keterangan
1	BaseTimeSeries	String name List<Tick> ticks	Timeseries series	Membuat object berisi kumpulan data historis saham
2	ClosePrice Indicator	Timeseries series	ClosePrice Indicator	Membuat object berisi kumpulan data historis harga penutupan saham

**Tabel 2.1** Fungsi pada TA4J yang digunakan

No	Nama Method	Input	Output	Keterangan
3	SMAIndicator	Indicator<Decimal> indicator Integer timeFrame	SMAIndicator	Membuat object berisi kumpulan data historis saham
4	WMAIndicator	Indicator<Decimal> indicator Integer timeFrame	WMAIndicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator WMA
5	Stochastic Oscillator K Indicator	Indicator<Decimal> indicator Integer timeFrame	Stochastic Oscillator K Indicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator STC%K
6	Stochastic Oscillator D Indicator	Stochastic Oscillator K Indicator	Stochastic Oscillator D Indicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator STC%D
7	Williams%R Indicator	Indicator<Decimal> indicator Integer timeFrame	Williams%R	Membuat object berisi kumpulan nilai hasil kalkulasi indikator Williams%R
8	MACDIndicator	Indicator<Decimal> indicator Integer shortTimeFrame Integer longTimeFrame	MACDIndicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator MACD
9	Accumulation Distribution Indicator	Timeseries series	Accumulation Distribution Indicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator AD
10	RSIIIndicator	Indicator<Decimal> indicator Integer timeFrame	RSIIIndicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator RSI

**Tabel 2.1** Fungsi pada TA4J yang digunakan

No	Nama Method	Input	Output	Keterangan
11	CCIIndicator	Indicator<Decimal> indicator Integer timeFrame	CCIIndicator	Membuat object berisi kumpulan nilai hasil kalkulasi indikator CCI
12	getClosePrice	-	Decimal	<i>Method</i> untuk mengambil harga penutupan saham dari object tick
13	getVolume	-	Decimal	Method untuk mengambil volume saham dari object tick
14	getOpenPrice	-	Decimal	Method untuk mengambil harga pembukaan saham dari object tick
15	getMaxPrice	-	Decimal	Method untuk mengambil harga terbesar saham dari object tick
16	getMinPrice	-	Decimal	Method untuk mengambil harga terkecil saham dari object tick
17	getValue	Integer index	T	Method untuk mengambil nilai indikator dari object indikator

### 2.8 Tinjauan Pustaka

Pada sub-bab ini akan dibahas mengenai refensi yang terkait dengan prediksi harga saham.

#### 2.8.1 Saham

Saham adalah surat berharga yang mewakili kepemilikan dari sebuah perusahaan dan menandakan klaim sebagian keuntungan dan asset dari sebuah perusahaan. Memprediksi pergerakan harga saham sangat sulit karena banyak sekali faktor-faktor yang mempengaruhi. Salah satu teori yang menjadi alasan sulitnya memprediksi pergerakan harga saham adalah *Efficient Market Hypothesis*[18]. Hipotesis tersebut menjelaskan bahwa harga sebuah sekuritas tidak akan terpengaruh apabila sebuah kumpulan informasi digunakan untuk menentukan posisi harga oleh seluruh partisipan. Dengan begitu, maka akan mustahil untuk memperoleh keuntungan dengan menggunakan kumpulan informasi tersebut.

*Efficient Market Hypothesis* memiliki beberapa bentuk efisiensi yang berbeda[18]:

##### 1. Weak Form

Bentuk ini menegaskan bahwa harga saham dapat merefleksikan seluruh informasi yang terdapat pada kumpulan data historis. Strategi investasi dengan menggunakan analisis pola harga terdahulu(analisa teknis) tidak dapat digunakan untuk meraih keuntungan. Bentuk ini disebut juga sebagai “*Random Walk Hypothesis*”.

##### 2. Semi-strong Form

Bentuk ini menegaskan bahwa harga saham selain merefleksikan informasi dari kumpulan data historis, harga juga merefleksikan informasi yang relevan terhadap sekuritas perusahaan. Strategi investasi dengan menggunakan analisis dari laporan posisi keuangan, laporan laba rugi, pengumuman tentang perubahan dividen atau pembagian saham, dan berbagai informasi publik tentang perusahaan tersebut (analisa fundamental) tidak dapat digunakan untuk meraih keuntungan.

##### 3. Strong Form

Bentuk ini menegaskan bahwa semua informasi yang diketahui oleh seluruh partisipan pasar tentang sebuah perusahaan akan terefleksikan seluruhnya pada harga saham. Walaupun seseorang memiliki hak untuk memperoleh sebuah informasi khusus, orang tersebut tidak dapat menggunakan informasi tersebut untuk meraih keuntungan yang besar.

Banyak yang memperdebatkan kebenaran teori tersebut karena jika teori tersebut benar, maka semua teknik analisis saham tidak akan berguna. Perilaku pasar saham yang sangat acak dan tidak dapat diprediksi hanya berlaku apabila tidak dapat memahami aturan yang berlaku. Dengan melakukan analisa saham secara tepat, maka seseorang dapat memahami bahwa perilaku acak dari sebuah saham tidak selalu terjadi setiap saat.

### 2.8.2 Analisis Saham

Beberapa teknik analisis yang dapat dilakukan untuk menganalisa pergerakan sebuah saham, yaitu analisis teknis dan analisis fundamental. Kedua analisis tersebut mempunyai tujuan yang sama, untuk memperkirakan arah pergerakan harga sebuah saham. Namun pendekatan yang dilakukan berbeda. Analisa fundamental mempelajari apa yang menyebabkan pergerakan sebuah saham, sementara analisa teknis mempelajari efek dari pergerakan sebuah saham[1].

Analisa teknis berkembang dari prinsip-prinsip tertentu yang berlaku umum. Prinsip-prinsip ini berasal dari Dow Theory. Dow Theory dihasilkan dari rangkaian artikel yang ditulis oleh Charles Dow di harian The Wall Street Journal antara tahun 1900 hingga 1902 dan awalnya hanya digunakan sebagai barometer untuk mengetahui kondisi bisnis secara umum[3]. Pada masa kini, Dow Theory digunakan sebagai dasar dalam pengembangan analisa teknis.

Dow Theory memiliki 6 asumsi yang menjadi dasar dari analisis teknis:

#### 1. *Market Action Discounts Everything*

Harga saham merefleksikan semua informasi yang ada padanya. Ketika informasi baru muncul, maka pelaku pasar dengan cepat akan menerjemahkan informasi dalam bentuk harga. Dengan demikian, secara umum pasar akan mendiskon dan merefleksikan semua informasi yang dimiliki oleh para pelaku pasar.

#### 2. *The Market Is Comprised of Three Trends.*

Ada tiga tren yang terjadi di pasar, yaitu *Primary Trend*, *Secondary Trend*, dan *Minor Trend*. Pada *Primary Trend*, pasar bisa bergerak naik (*bullish*) atau turun (*bearish*). *Primary Trend* dapat berlangsung selama lebih dari 1 tahun hingga beberapa tahun. *Secondary trend* adalah koreksi terhadap *Primary Trend*. Koreksi ini berlangsung selama satu hingga tiga bulan. *Minor trend* adalah pergerakan harga dalam jangka pendek yang berlangsung dalam 1 hari sampai tiga minggu. Dow Theory mengatakan bahwa *Primary Trend* dan *Secondary Trend* tidak dapat dimanipulasi, sedangkan *Minor trend* memiliki kemungkinan dimanipulasi. Karena itu *Minor Trend* dapat memberikan petunjuk yang salah.

#### 3. *Primary Trend Have Three Phases.*

Menurut Dow Theory, *Primary Trend* memiliki tiga fase. Fase pertama dimana investor tertentu melakukan pembelian dalam jumlah yang sangat banyak untuk mengantisipasi pemulihan ekonomi dan pertumbuhan jangka panjang. Fase kedua ditandai oleh para investor yang mulai mengakumulasi saham seiring dengan meningkatnya pendapatan perusahaan dan meningkatnya kondisi ekonomi. Fase ketiga ditandai dengan pendapatan perusahaan yang mencapai puncak dan kondisi ekonomi makro yang sangat baik. Investor retail mulai merasa aman dalam berpartisipasi dalam investasi saham. Mereka yakin saham akan bisa tumbuh lebih tinggi, sehingga mereka membeli lebih banyak saham. Saat fase ini terjadi euforia saham, investor publik membeli saham jauh lebih banyak lagi. Para investor yang melakukan pembelian di fase pertama akan mulai menjual sahamnya untuk mengantisipasi penurunan harga saham.

### 4. *The Averages Must Confirm Each Other.*

Harga rata-rata indeks saham gabungan harus saling mengkonfirmasi agar tren yang *valid* terjadi. Harga rata-rata indeks saham gabungan harus melebihi puncak *Secondary Trend* sebelumnya.

### 5. *The Volume Confirms the Trend.*

Volume hanya digunakan untuk mengkonfirmasi situasi yang tidak pasti. Volume harus meningkat sesuai pergerakan *Primary trend*. Jika *Primary trend* bersifat turun maka volume harus meningkat di saat pasar jatuh. Jika *Primary trend* bersifat meningkat maka volume harus meningkat di saat pasar bergerak naik.

### 6. *A Trend Remains Intact Until it Gives a Definite Reversal Signal.*

*Uptrend* ditunjukkan oleh serangkaian pergerakan harga yang berhasil mencapai puncak lebih tinggi. Agar terjadi pembalikan arah, harga harus setidaknya gagal mencapai kembali puncak yang sudah terjadi sebelumnya (terbentuk *lower high*), dan terjadi penurunan yang lebih dalam dari koreksi sebelumnya(terbentuk *lower low*). Ketika terjadi pembalikan arah di *Primary Trend* ditunjukkan bersama-sama oleh indeks gabungan, kemungkinan terjadinya pembalikan arah dan berlanjut ke tren tersebut semakin besar. Bagaimanapun, semakin lama suatu tren berlangsung kemungkinan suatu tren tetap berlanjut semakin kecil.

Masalah yang sering terjadi adalah hasil analisa teknis dan analisa fundamental sering menghasilkan hasil yang berbeda apabila digunakan secara bersamaan. Hal tersebut terjadi karena adanya pergerakan harga yang sangat sedikit dan nilai informasi yang dibawa oleh harga saham pada saat itu sudah berkurang. Bila dilihat dari sisi analisa fundamental, perubahan yang terjadi sangat sedikit atau tidak terlihat adanya perubahan. Sementara dari sisi analisa teknis, perubahan yang terjadi akan terlihat dengan *chart* yang digunakan untuk menganalisis. Dampak dari masalah ini adalah telatnya reaksi investor dalam menanggapi perubahan situasi pada pasar saham dan dapat menyebabkan kerugian yang cukup besar.

Analisis saham menggunakan analisa teknis dapat dilakukan jika pergerakan harga saham mengikuti tren saham. Pergerakan tren saham akan bergerak secara koturn hingga menunjukkan tanda bahwa tren tersebut akan bergerak ke arah sebaliknya. Pergerakan berbalik arah ini terjadi karena adanya perubahan *supply* dan *demand* pasar[19]. Ketika tren saham sedang bergerak naik maka yang terjadi adalah semakin besar *supply* atau orang yang bersedia untuk menjual sahamnya dan semakin sedikit *demand* atau orang yang bersedia membeli saham tersebut. Ketika tren saham bergerak turun maka yang terjadi adalah semakin sedikit *supply* dan semakin banyak *demand*.

Jika harga saham bergerak tidak mengikuti tren saham maka saham tersebut dianggap sedang mengalami *Unusual Market Activity*(UMA). Karena pergerakan harga yang tidak sesuai dengan tren maka analisa teknis tidak dapat digunakan pada saham yang mengalami *Unusual Market Activity*. Untuk mengetahui sebuah saham tidak mengalami UMA, biasanya setiap aktifitas *trading* yang *valid* akan disertai dengan jumlah volume *trading* yang cukup besar.

### 2.8.3 Indikator Teknis Saham

Ada beberapa indikator teknis yang dapat digunakan untuk memprediksi pergerakan saham:

#### 2.8.3.1 Moving Average

*Moving average* dapat digunakan sebagai alat bantu analisis saham dalam berbagai hal, yaitu[3]:

1. Mengidentifikasi tren saham.
2. Mengetahui pembalikan arah tren saham.
3. Menentukan tingkat *support/resistance*.

*Simple Moving Average(SMA)* dan *Weighted Moving Average(WMA)* termasuk salah satu varian dari moving average. SMA hanya menggunakan rata-rata harga biasa. WMA menggunakan rata-rata harga yang disertai dengan sistem pembobotan. Setiap data dari periode yang berbeda akan diberi pembobotan yang berbeda sehingga menghasilkan nilai rata-rata yang berbeda. WMA akan lebih sensitif terhadap perubahan harga dibandingkan dengan SMA.

Untuk mengetahui tren dapat dilihat dari posisi MA terhadap harga. Bila harga memotong MA dari atas ke bawah maka tren menjadi *bearish* dan memberikan sinyal untuk menjual. Bila harga memotong MA dari bawah ke atas maka tren menjadi *bullish* dan memberikan sinyal untuk membeli.

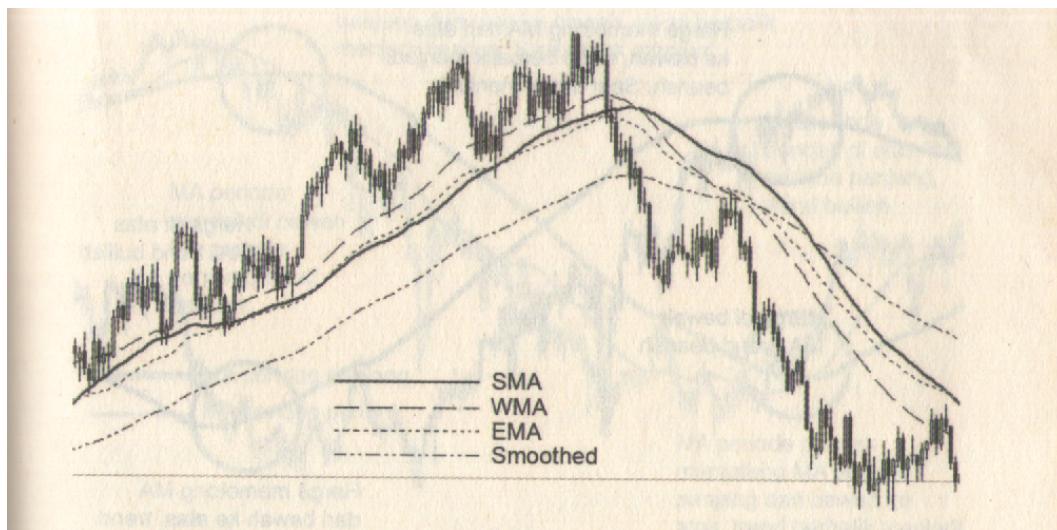
Indikator SMA dan WMA dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$\text{SimpleMA}(n) = \frac{C_1 + C_{(t-1)} + C_{(t-2)} + \dots + C_{t-(n-1)}}{n} \quad (2 . 7)$$

$$\text{WeightedMA}(n) = \frac{(n)C_t + (n-1)C_{t-1} + (n-2)C_{t-2} + \dots + (n-(n-1))C_{t-(n-1)}}{n + (n-1) + (n-2) + \dots + 1} \quad (2 . 8)$$

Dimana:

$C_t$  = Harga pada waktu *closing*



**Gambar 2.1** Contoh Grafik Indikator *Moving Average* [3]

### 2.8.3.2 Stochastic Oscillator

*Stochastic* dikembangkan oleh George C. Lane pada tahun 1950[3]. *Stochastic Osciliator* digunakan untuk menunjukkan lokasi harga penutupan terakhir dibandingkan dengan range harga terendah/tertinggi selama periode waktu tertentu. Ada tiga macam tipe *Stochastic Osciliator* yaitu *Fast*, *Slow*, dan *Full*. Namun jenis yang sering digunakan oleh trader adalah *Stochastic Slow* karena grafik yang lebih mudah untuk dibaca. Parameter yang disarankan untuk menggunakan indikator *Stochastic* adalah (5, 3) [3].

Ada dua prinsip yang perlu diketahui ketika menggunakan *Stochastic Slow* :

1. Nilai *Stochastic* di atas 80 dikatakan *overbought* dan kemungkinan harga akan segera turun.  
Nilai *Stochastic* di bawah 20 dikatakan *oversold* dan kemungkinan harga akan segera naik.  
Sinyal kadang tidak 100% tepat. Misalnya jika nilai *Stochastic* sudah di atas 80, kemungkinan masih bisa akan naik lagi. Begitu juga sebaliknya.
2. Sinyal jual dan beli dapat dilihat dari garis %K dan %D. Jika %K memotong %D ke atas maka sinyal beli. Jika %K memotong %D ke bawah maka sinyal jual.

Indikator *Stochastic* %K dan %D dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$\text{Stochastic \%K} = \frac{C_t - LL_n}{HH_n - LL_n} * 100 \quad (2 . 9)$$

$$Stochastic \%D = \frac{\sum_{i=0}^3 K_{t-i}}{n} \% \quad (2 . 10)$$

Dimana:

$C_t$  = Harga *closing* pada waktu  $t$

$LL_n$  = Harga terendah selama periode waktu  $n$

$HH_n$  = Harga tertinggi selama periode waktu  $n$



**Gambar 2.2** Contoh Grafik Indikator STC%K dan STC%D[3]

### 2.8.3.3 Williams' Percent Range

Indikator *Williams' Percent Range*(W%R) adalah indikator yang berfungsi untuk menunjukkan apakah suatu harga sudah *overbought* atau *oversold*[3]. Bila indikator menunjukkan nilai antara 80% sampai 100% maka dapat dikatakan *oversold*, kemungkinan harga akan bergerak naik. Bila indikator menunjukkan nilai antara 0% sampai 20% maka dapat dikatakan *overbought*, kemungkinan harga akan bergerak turun.

Indikator *William's %R* dapat dihitung dengan menggunakan rumus berikut[5], [19]:

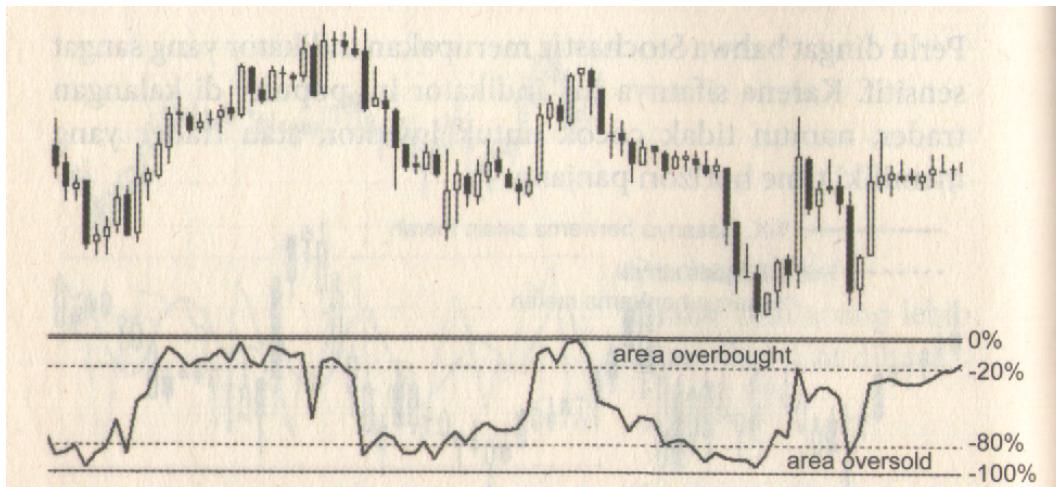
$$Williams' \%R = \frac{HH_n - C_t}{HH_n - LL_n} * 100 \quad (2 . 11)$$

Dimana:

$C_t$  = Harga *closing* pada waktu  $t$

$LL_n$  = Harga terendah selama periode waktu  $n$

$HH_n$  = Harga terendah selama periode waktu  $n$



Gambar 2.3 Contoh Grafik Indikator Williams' %R[3]

#### 2.8.3.4 Moving Average Convergence Divergence

Indikator *Moving Average Convergence Divergence*(MACD) digunakan untuk menunjukkan tren saham yang sedang terjadi[3]. Jika nilai MACD positif(di atas nol), maka tren sedang *bullish*. Sedangkan jika nilai MACD negatif(di bawah nol), maka tren sedang *bearish*. Parameter yang disarankan untuk menggunakan indikator MACD adalah (12,26) [16].

Indikator MACD dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$EMA(k)_t = EMA(k)_{t-1} + \left( \frac{2}{k+1} \right) * (C_t - EMA(k)_{t-1}) \quad (2 . 12)$$

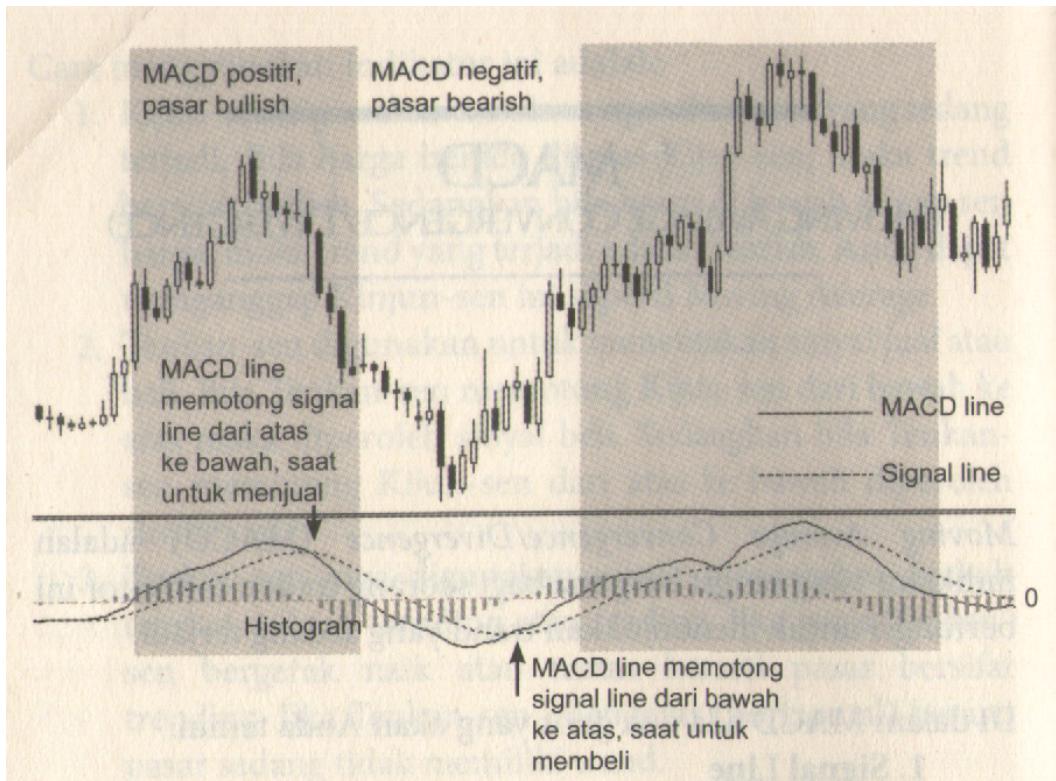
$$MACD = EMA(x_1)_t - EMA(x_2)_t \quad (2 . 13)$$

Dimana:

$k$  = Waktu periode EMA

$x_1$  = Periode EMA cepat

$x_2$  = Periode EMA lambat



**Gambar 2.4** Contoh Grafik Indikator MACD[3]

#### 2.8.3.5 Accumulation/Distribution Oscillator

Indikator *Accumulation/Distribution*(A/D) adalah indikator momentum yang berkaitan dengan perubahan harga dan volume[3]. Bila nilai A/D naik maka harga akan bergerak naik, bila nilai A/D turun maka harga akan bergerak turun.

Bila terdapat anomali, misalnya nilai A/D naik namun harga malah bergerak turun maka diperkirakan harga akan bergerak keatas dengan segera.

Indikator A/D dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$\text{Money Flow Multiplier} = \frac{(C_t - L_t) - (H_t - C_t)}{H_t - L_t} \quad (2 . 14)$$

$$Money Flow Volume = Money Flow Multiplier * V_t \quad (2 . 15)$$

$$A/D Line = Previous A/D Line + Current Money Flow Volume \quad (2 . 16)$$

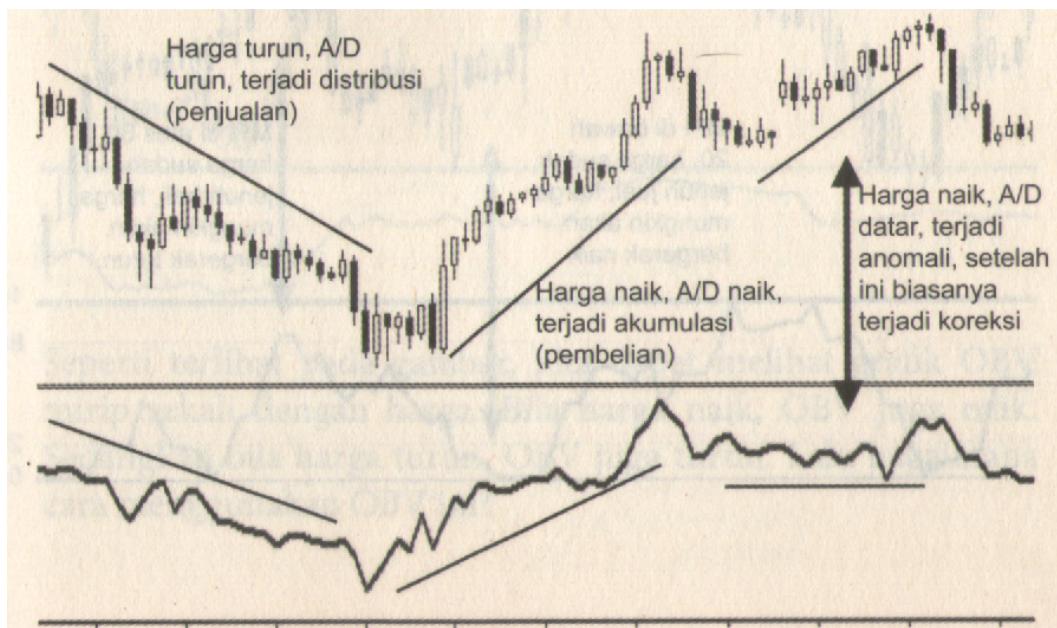
Dimana:

$C_t$  = Harga *closing* pada waktu  $t$

$L_t$  = Harga terendah pada waktu  $t$

$H_t$  = Harga terendah pada waktu  $t$

$V_t$  = Volume saham pada waktu  $t$



Gambar 2.5 Contoh Grafik Indikator A/D[3]

#### 2.8.3.6 Relative Strength Index

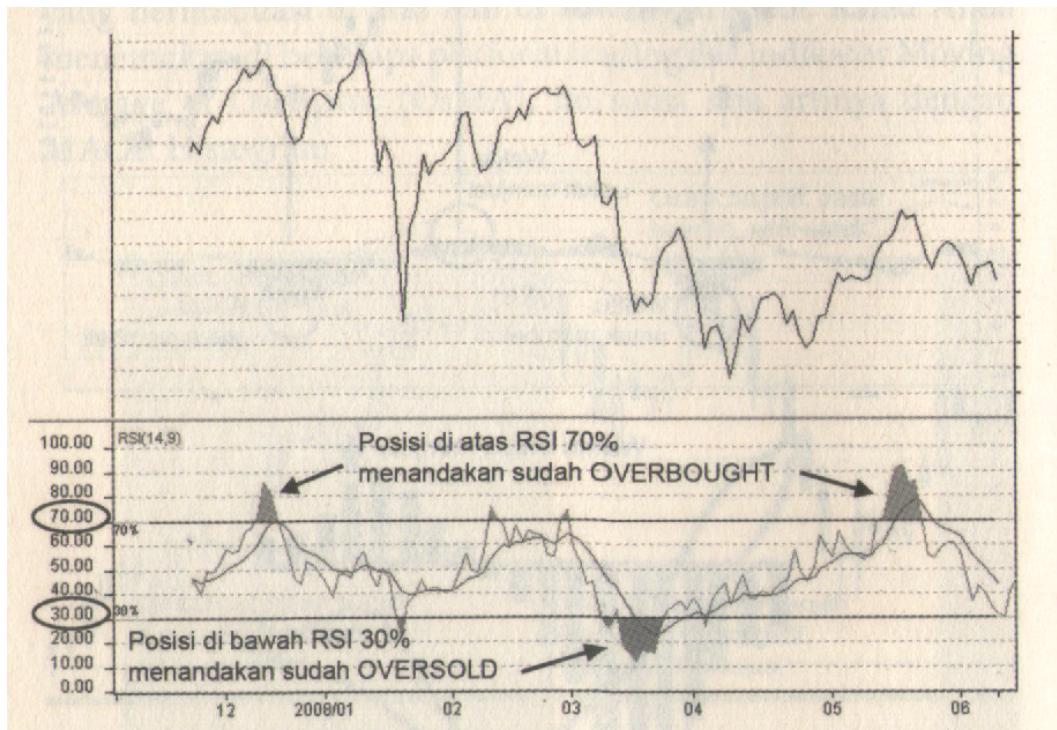
Indikator *Relative Strength Index*(RSI) digunakan untuk mengidentifikasi apakah sebuah saham sudah terlalu banyak dibeli(*overbought*) atau sudah terlalu banyak dijual(*oversold*)[3]. Indikator RSI mempunyai rentang nilai dari 0 hingga 100. Apabila nilai RSI lebih besar dari 70 maka saham tersebut berada dalam kondisi *overbought* dan tren saham

dipastikan akan menurun, kemudian bila nilai RSI lebih kecil dari 30 maka saham tersebut berada dalam kondisi *oversold* dan tren saham dipastikan akan meningkat.

Indikator RSI dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$RSI = 100 - \frac{100}{1 + RS} \quad (2 . 17)$$

$$RS = Average\ Gain - Average\ Loss \quad (2 . 18)$$



**Gambar 2.6** Contoh Grafik Indikator RSI[3]

#### 2.8.3.7 Community Channel Index

Indikator *Commodity Channel Index*(CCI) dikembangkan oleh Donald Lambert[3]. CCI digunakan untuk mengidentifikasi tingkat *overbought* dan *oversold*. Nilai indikator CCI akan berfluktuasi dari nilai -100 hingga 100. Bila menunjukkan nilai yang tinggi menunjukkan sinyal tren untuk menjual, dan bila menunjukkan nilai yang rendah menunjukkan sinyal tren untuk membeli. Bila nilai CCI berada di atas 100 dikatakan *overbought*. Harga sudah jenuh beli dan berkemungkinan harga akan segera berbalik turun. Bila nilai CCI berada di bawah -100

dikatakan *oversold*. Harga sudah jenuh jual dan berkemungkinan harga akan segera berbalik naik.

Indikator CCI dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$TP_t = \frac{H_t + L_t + C_t}{3} \quad (2 . 19)$$

$$Average\ TP_t = \frac{\sum_{i=1}^n TP_i}{n} \quad (2 . 20)$$

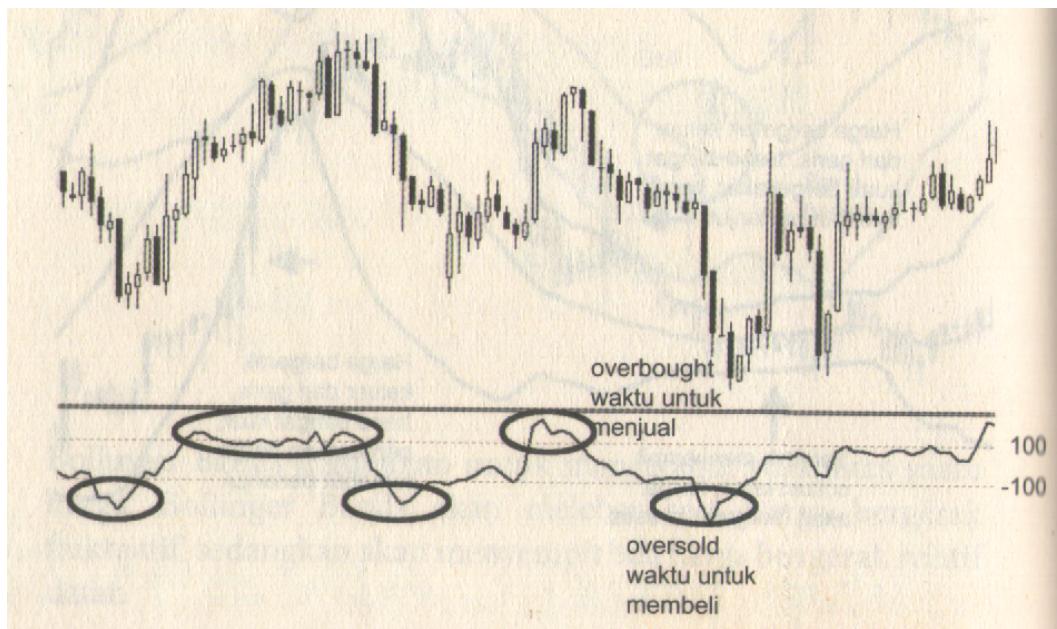
$$D_t = \frac{(\sum_{i=1}^n |Average\ TP_t - TP_i|)}{n} \quad (2 . 21)$$

$$CCI = \frac{TP_t - Average\ TP_t}{0.015D_t} \quad (2 . 22)$$

Dimana:

$TP_t$  = Typical Price

$D_t$  = Deviation



**Gambar 2.7** Contoh Grafik Indikator CCI[3]

#### 2.8.3.8 Momentum

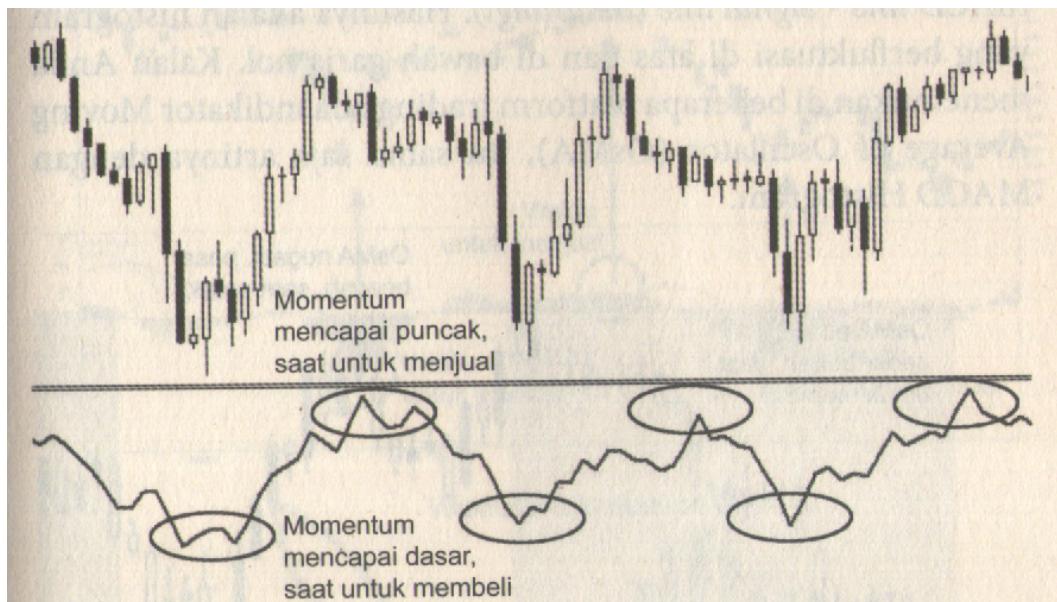
Indikator *Momentum* adalah indikator yang digunakan untuk mengukur perubahan harga selama periode waktu tertentu[3]. Cara menggunakan indikator *Momentum* sama seperti menggunakan indikator RSI. Jika nilai indikator *Momentum* positif maka menunjukkan tren saham sedang meningkat dan jika nilai indikator *Momentum* negatif maka menunjukkan tren saham sedang menurun.

Indikator *Momentum* dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$\text{Momentum} = C_t - C_{t-(n-1)} \quad (2 . 23)$$

Dimana:

$C_t$  = Harga *closing* pada waktu  $t$



**Gambar 2.8** Contoh Grafik Indikator *Momentum*[3]

#### 2.8.3.9 Alligator Indicator

Indikator ini diciptakan oleh Bill Williams. Cara kerja indikator ini seperti mulut buaya yang sedang terbuka. Bila mulut buaya sedang terbuka ke atas berarti saham sedang mengalami *uptren*, dan bila mulut buaya sedang terbuka ke bawah maka saham sedang mengalami *downtrend*.

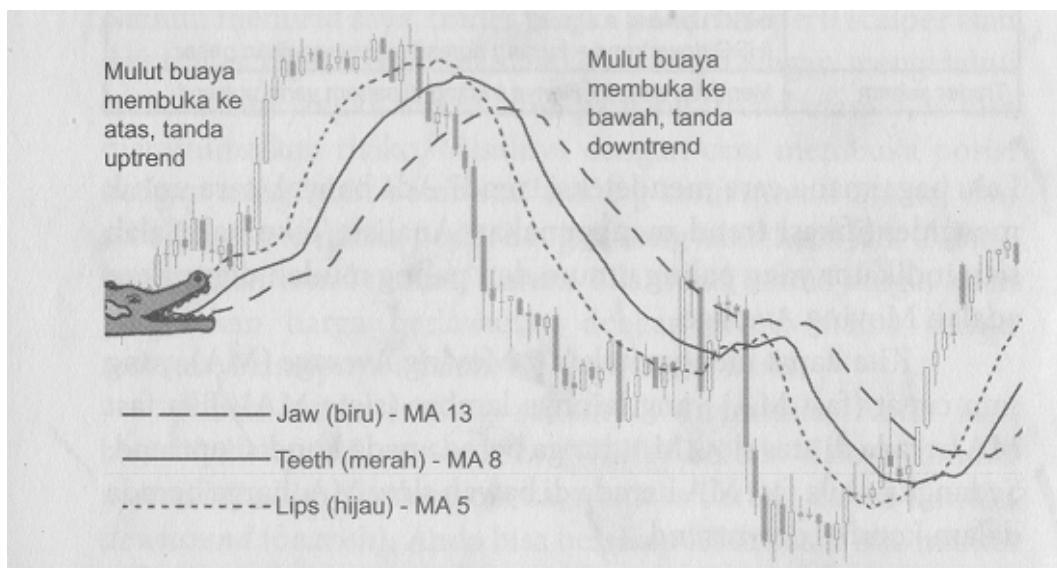
Indikator *Alligator* menggunakan 3 SMMA(*Smoothed Moving Average*) untuk merepresentasikan rahang, gigi, dan bibir buaya pada chart. Kekurangan dari indikator *Alligator* adalah kurang bagus dalam mendeteksi tren *sideways*. Ketika kondisi sedang *sideways*, banyak *false signal* yang muncul sehingga membuat indikator tidak dapat mendeteksi tren dengan baik. Parameter yang disarankan untuk menggunakan indikator *Alligator* adalah (13,8,5)[3].

Indikator *Alligator* dapat dihitung dengan menggunakan rumus berikut[5], [19]:

$$SMMA(N) = \frac{((SMMA_{t-1} * (N - 1)) + C_t)}{N} \quad (2 . 24)$$

Dimana:

$C_t$  = Harga *closing* pada waktu  $t$



**Gambar 2.9** Contoh Grafik Indikator *Alligator*[3]

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini menjelaskan mengenai analisis dan perancangan yang digunakan untuk implementasi. Untuk menguji keakuratan dari penelitian ini, maka penulis melakukan analisis dengan membuat kerangka pemikiran dan *flow chart*.

#### **3.1 Analisis Masalah**

Penelitian ini bertujuan untuk memprediksi pergerakan harga indeks saham berdasarkan tren dari saham tersebut dengan menggunakan metode *Random Forest*. Seperti yang dijelaskan pada Bab I, pergerakan harga saham hampir tidak dapat bisa ditebak karena banyak sekali faktor yang mempengaruhi pergerakannya. Namun, dengan menganalisa pergerakan harga saham maka terlihat arah pergerakan harga saham. Karena jumlah indikator teknis sangat banyak maka dengan menggunakan *Random Forest* akan membantu mengklasifikasikan data saham untuk menentukan pergerakan arah harga saham berdasarkan hasil analisis indikator teknis.

Masukan yang digunakan pada penelitian ini berupa data historis saham dengan format CSV. Data berisi harga pembukaan, harga penutupan, harga tertinggi, harga terendah, dan volume saham pada setiap harinya. Data didapat dari finance.google.com secara gratis.

Data masukan akan diproses melalui tahap *feature extraction* terlebih dahulu. Tujuannya adalah untuk mengambil informasi-informasi tentang pergerakan saham dari setiap indikator teknis berdasarkan data masukan. Pada tahap *feature extraction* juga akan dilakukan proses diskritasi data. Proses ini mengubah setiap nilai fitur data yang sebelumnya numerik menjadi bentuk diskrit.

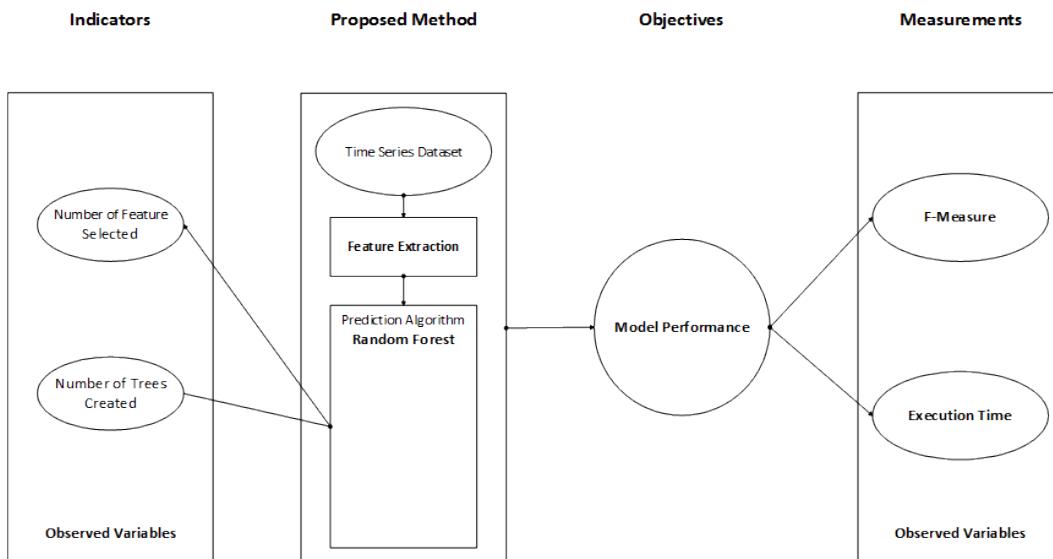
Setelah itu, data akan digunakan sebagai data training untuk membangun kumpulan pohon keputusan. Setiap pohon akan diberikan sebuah set data yang berbeda-beda agar dan fitur data akan dipilih secara acak ketika melakukan percabangan pohon keputusan. Ketika kumpulan pohon keputusan sudah terbentuk, maka kumpulan pohon keputusan tersebut akan digunakan untuk mengklasifikasikan data menjadi 2 kelas yaitu harga naik dan harga turun.

Penelitian ini menggunakan 2 set data historis dari 2 perusahaan yang berbeda. Hasil klasifikasi tidak akan disimpan karena ketika data yang diprediksi sudah diketahui kelas yang sebenarnya maka untuk data training selanjutnya akan menggunakan data yang diprediksi sebelumnya dengan menggunakan kelas yang sebenarnya.

### 3.2 Kerangka Pemikiran

Berikut adalah kerangka pemikiran dari metode yang diusulkan untuk melakukan prediksi harga saham. Penelitian ini bertujuan untuk menguji akurasi dan waktu prediksi dengan metode *Random Forest* untuk tahap klasifikasinya.

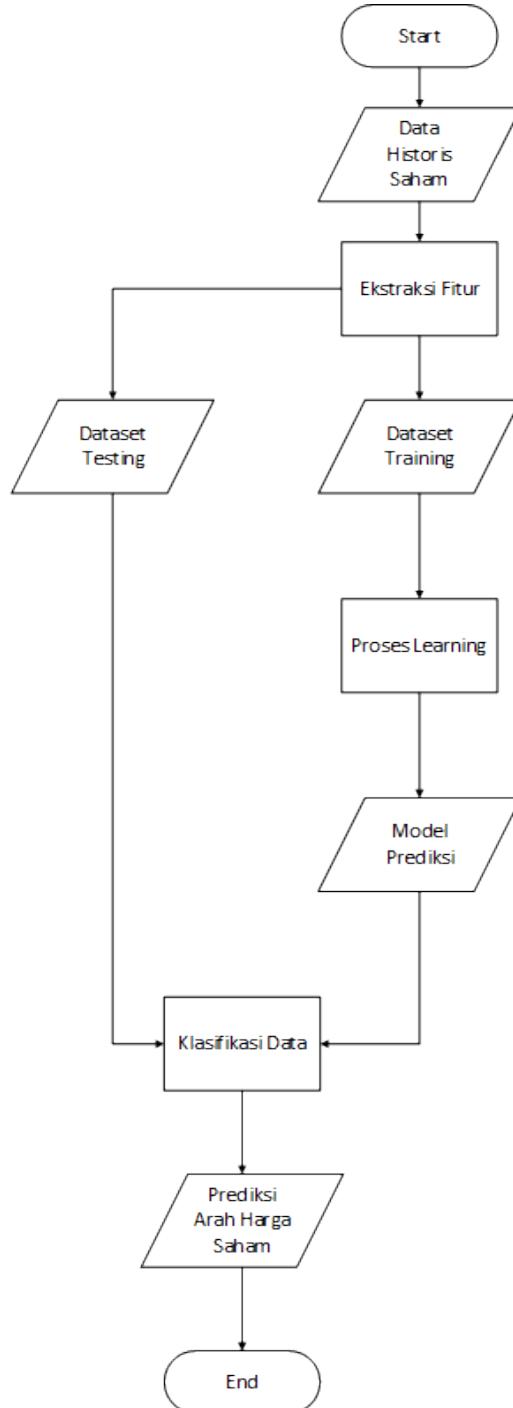
Dengan keterangan sebagai berikut:



**Gambar 3.1** Diagram Kerangka Pemikiran

1. *Indikator*: Faktor yang mempengaruhi hasil pengujian metode utama. Indikator pada proses klasifikasi adalah Jumlah pohon keputusan dan jumlah fitur yang dipilih acak.
2. *Proposed Method*: Metode yang diusulkan pada penelitian ini:
  - (a) *Feature Extraction*: Tahap mengekstrasi informasi-informasi dari data yang kemudian akan digunakan untuk diproses pada metode utama. Tahap ini akan melakukan proses ekstraksi indikator teknis dan diskritasi data.
  - (b) *Prediction Algoritm*: Tahap dimana metode utama melakukan proses klasifikasi untuk menentukan arah harga saham.
3. *Objective*: Penelitian ini akan menguji performa metode yang digunakan pada penelitian ini.
4. *Measurements*: Penelitian ini akan menggunakan *f-measure* dan waktu eksekusi metode sebagai indikator pengukuran performa metode.

### 3.3 Tahap Analisis Urutan Proses Umum



**Gambar 3.2** Diagram Proses Umum

Pada penelitian ini, dataset *training* dan dataset *testing* adalah 2 dataset yang identik. Namun saat proses testing, data yang ditest tidak akan pernah sama. Data testing akan selalu lebih baru daripada data training jika dilihat dari tanggal datanya.

Uraian dari urutan proses umum gambar 3.2:

1. Data masukan yang akan diberikan adalah data historis saham dengan format CSV, data didapatkan dari situs [finance.google.com](http://finance.google.com).

2. Proses ekstraksi fitur akan mengekstraksi informasi dari setiap indikator teknis saham. Proses ekstraksi fitur akan dilakukan oleh library TA4J. Pada tahap ini juga dilakukan proses diskritasi data. Diskritasi dilakukan setelah ekstraksi fitur selesai, kemudian nilai-nilai fitur data akan diubah menjadi bentuk diskrit. Proses diskritasi akan mengikuti perilaku masing-masing indikator saham.
3. Pada proses *learning*, setiap pohon keputusan akan dibentuk dengan dataset yang berbeda dan ketika merancang pohon keputusan fitur data yang digunakan sebagai pembagi cabang pohon akan dipilih secara anak. Tujuannya agar tidak ada pohon keputusan lain yang memiliki struktur yang serupa.
4. Proses klasifikasi dilakukan dengan memasukan data ke setiap pohon keputusan yang telah dibuat. Setiap pohon akan menghasilkan hasil klasifikasi yang berbeda. Hasil akhir klasifikasi akan diambil berdasarkan voting dari seluruh pohon keputusan.

#### 3.3.1 Analisis Data

Data set yang digunakan adalah data historis saham yang setiap datanya memiliki atribut tanggal data, harga pembukaan saham, harga penutupan saham, harga tertinggi saham pada tanggal tersebut, harga terendah pada tanggal tersebut, dan volume saham. Periode data yang digunakan adalah sepanjang 6 tahun dalam rentang waktu tahun 2010-2015.

#### 3.3.2 Ekstraksi Fitur

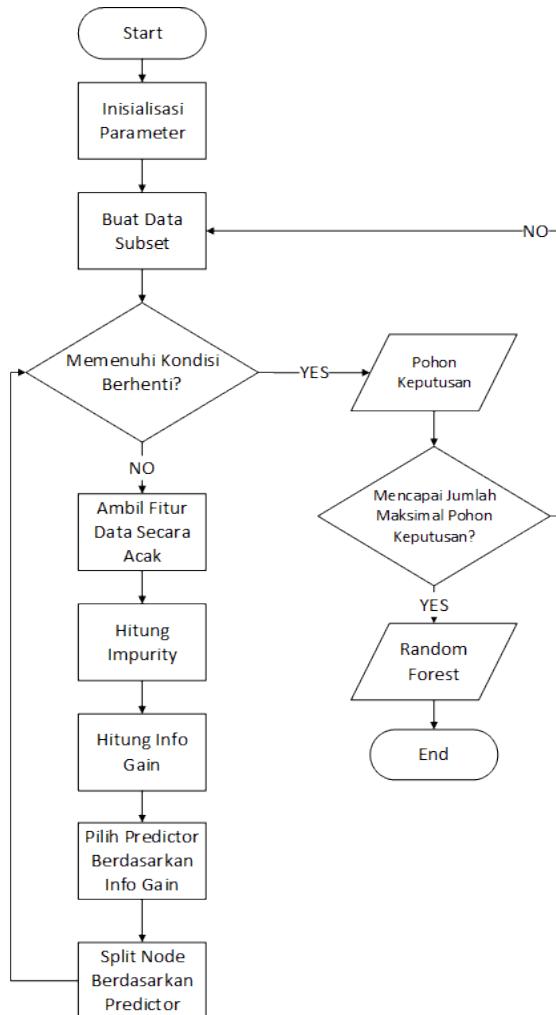
Beberapa indikator teknis yang akan diekstrak dari data adalah SMA, WMA, *Momentum*, *Stochastic K%*, *Stochastic D%*, RSI, MACD, *William's R%*, A/D *Oscillator*, CCI, dan *alligator*. Setiap indikator memiliki perilaku dan fungsi yang berbeda. SMA dan WMA yang sama-sama merata-ratakan harga namun WMA lebih sensitif terhadap perubahan harga dibandingkan dengan SMA. A/D digunakan untuk mengetahui momentum antara perubahan harga dan volume. RSI, CCI, *Stochastic%K*, *Stochastic%D*, dan *Williams'%R* digunakan untuk mengidentifikasi apakah harga sudah overbought atau oversold dari perspektif yang berbeda. MACD digunakan untuk mengidentifikasi trend yang sedang terjadi. Momentum digunakan untuk mengukur perubahan harga selama periode waktu tertentu.

Proses diskritasi dilakukan setelah data telah diekstraksi dari dataset. Proses diskritasi mengubah bentuk nilai atribut dari numerik menjadi diskrit[+1, -1]. Proses diskritasi berfungsi untuk menjaga informasi yang dibawa oleh setiap fitur data ketika digunakan sebagai input bagi metode prediksi dan dapat meningkatkan akurasi dari metode prediksi.

#### 3.3.3 Proses *Learning*

Parameter yang diterapkan adalah jumlah total iterasi untuk membangun pohon keputusan sebanyak  $n$ . Data *training* akan diambil sebanyak 40 hari kebelakang dari hari dimana pergerakan harga selanjutnya akan diprediksi. Tujuan mengambil data sebanyak 40

hari untuk data *training* adalah untuk menangkap pola yang dihasilkan oleh indikator-indikator teknis dan bila data *training* terlalu sedikit maka pola yang dihasilkan oleh indikator-indikator teknis tidak dapat dianalisa oleh metode prediksi. Untuk fase *training*, setiap pohon kesempatan akan dibuatkan subset data yang berbeda dari data *training*. Setiap data pada subset dipilih secara acak dari data set *training* dan fitur-fitur data yang digunakan pada subset tersebut dipilih secara acak sebanyak  $\sqrt{m}$  dari total  $m$  fitur data *training* pada setiap proses pembagian *node*. Setiap subset data akan digunakan untuk membangun 1 pohon keputusan.



**Gambar 3.3** Diagram Proses Learning

### 3.3.4 Klasifikasi

Setiap data pada data set testing akan diujicobakan kepada setiap pohon keputusan yang telah dibuat pada fase *training*. Hasil dari fase *testing* untuk setiap data adalah label dengan jumlah terbanyak yang dihasilkan dari seluruh pohon keputusan.

### 3.4 Perancangan Kasus

Pada bagian ini terdapat analisis dengan melakukan perhitungan manual metode-metode yang digunakan untuk klasifikasi data saham.

#### 3.4.1 Feature Extraction

Diberikan sebuah data historis saham dari sebuah perusahaan. Data tersebut akan digunakan untuk proses training model *Random Forest*. Fitur-fitur yang akan digunakan oleh model *Random Forest* akan diekstrak dari data set dengan menggunakan 10 indikator teknis. Analisis akan dilakukan menggunakan parameter selama 10 hari untuk semua indikator data agar dapat menangkap trend yang terjadi berdasarkan kejadian pasar selama 2 minggu sebelumnya(kecuali MACD menggunakan parameter selama 12 dan 26 hari dan indikator *alligator* menggunakan parameter 5,8,13 berdasarkan parameter yang disarankan). Untuk menentukan arah harga pada setiap data, akan ditentukan berdasarkan perbandingan harga pada hari tersebut dan hari sebelumnya. Jika lebih besar dari hari sebelumnya maka dinyatakan arah harga sedang naik, jika lebih kecil maka dinyatakan arah harga sedang turun.

**Tabel 3.1** Contoh Data Historis Saham

No	Date	Open	High	Low	Close	Volume
1	23-Sep-11	14900	15350	14650	15350	1207500
2	26-Sep-11	15500	15600	14650	15200	2554000
3	27-Sep-11	15400	15950	15400	15600	1319500
4	28-Sep-11	15500	15850	15450	15600	1934500
5	29-Sep-11	15300	16500	15300	15700	2209500
6	30-Sep-11	15700	16500	15700	16500	3422500
7	3-Oct-11	16500	16500	15350	16100	2423000
....	....	....	....	....	....	....
....	....	....	....	....	....	....
13	11-Oct-11	15750	15850	15550	15550	2350500
14	12-Oct-11	15500	16000	15500	16000	1322500
15	13-Oct-11	16200	16250	15850	15850	2175000
....	....	....	....	....	....	....

### III. ANALISIS DAN PERANCANGAN SISTEM

---

Contoh ekstraksi fitur dari data tanggal 13 Oktober 2011 :

1. *Simple Moving Average*

$$\text{SMA} = (15850 + 16000 + 15550 + 15600 + 15900 + 16050 + 15850 + 15750 + 16100 + 16500) / 10 = 15915$$

2. *Weighted Moving Average*

$$\begin{aligned}\text{WMA} = & ((10/55) * 15850) + ((9/55) * 16000) + ((8/55) * 15550) + ((7/55) * \\& 15600) + \\& ((6/55) * 15900) + ((5/55) * 16050) + ((4/55) * 15850) + ((3/55) * 15750) + \\& ((2/55) * 16100) + ((1/55) * 16500) \\= & 15838.18\end{aligned}$$

3. *Momentum*

$$\text{Momentum} = 15850 - 16500 = -650$$

4. *Stochastic %K*

$$\text{STC\%K} = (15850 - 15550) / (16500 - 15550) * 100 = 31.58$$

5. *Stochastic %D*

$$\text{STC\%D} = 78.95 / 3 = 26.32$$

6. *Relative Strength Index*

$$\begin{aligned}\text{RS} &= 130.20 / 139.72 = 0.93 \\ \text{RSI} &= 100 - 100 / (1 + 0.93) = 48.24\end{aligned}$$

7. *Moving Average Convergence Divergence (MACD)*

$$\text{MACD} = \text{EMA}(12) - \text{EMA}(26) = 15632.04 - 15989.21 = -357.17$$

8. *Williams %R*

$$\text{William's } \%R = (16500 - 15850) / (16500 - 15550) * 100 = 68.42$$

9. *Accumulation/Distribution Oscillator*

*Money Flow Multiplier* =

$$(((15850 - 15850) - (16250 - 15850))) / (16250 - 15850) = -1$$

*Money Flow Volume* =  $(-1) * (2175000 / 1000) = -2175$

$$\text{A/D} = 1600.31 + (-2175) = -574.69$$

10. *Commodity Channel Index*

$$\text{Typical Price} = (16250 + 15850 + 15850) / 3 = 15983.33$$

$$\text{Average Typical Price} = 159100 / 10 = 15910$$

*Deviation* =

$$(323.33 + 73.33 + 226.67 + 93.33 + 206.67 + 190 + 210 + 260 + 76.67 + 73.33) / 10 = 173.33$$

$$\text{CCI} = (15983.33 - 15910) / (0.015 * 246) = 28.21$$

11. *Indikator Alligator*

$$\text{Lips} = \text{SMMA}(3) = ((15818.47 * 2) + 158580) / 3 = 15828.98$$

$$\text{Teeth} = \text{SMMA}(5) = ((15834.77 * 4) + 158580) / 5 = 15837.82$$

$$\text{Jaw} = \text{SMMA}(8) = ((15892.85 * 7) + 15850) / 8 = 15892.85$$

$$\text{Tren} = 15892.85 > 15837.82 > 15828.98$$

Tren = Jaw > Teeth > Lips

Tren = Down

Tabel 3.2 Contoh hasil ekstraksi fitur data

No	Date	SMA	WMA	Momentum	STC%K	STC%D	RSI	MACD	%R	A/D	CCI	Alligator	Direction
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	30-Sep-11	15685	15705	450	100.00	56.76	56.89	-643.29	0.00	4080.51	137.57	Down	Up
2	3-Oct-11	15690	15780	50	73.33	70.39	50.12	-662.76	26.67	4817.94	70.26	Down	Up
3	4-Oct-11	15660	15791	-50	50.00	74.44	44.93	-630.00	50.00	6152.27	13.00	Down	Down
4	5-Oct-11	15665	15826	850	56.67	60.00	46.68	-592.03	43.33	6639.15	44.04	Down	Down
5	6-Oct-11	15770	15896	700	65.38	57.35	50.21	-511.66	34.62	5291.81	95.76	Down	Up
6	7-Oct-11	15825	15920	700	53.85	58.63	47.59	-447.70	46.15	3422.31	80.00	Down	Up
7	10-Oct-11	15865	15879	0	0.00	39.74	42.64	-416.42	100.00	2628.31	-62.50	Down	Down
8	11-Oct-11	15860	15821	-50	0.00	17.95	41.83	-391.16	100.00	277.81	-80.36	Down	Down
9	12-Oct-11	15900	15847	300	47.37	15.79	51.07	-407.93	52.63	1600.31	-24.09	Down	Down
10	13-Oct-11	15915	15838	-650	31.58	26.32	48.24	-357.17	68.42	-574.69	28.21	Down	Up
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Setelah melakukan proses ekstraksi fitur, kemudian data akan diubah bentuknya pada proses diskritasi nilai atribut data. Contoh proses diskritasi nilai fitur data:

1. *Simple Moving Average*

$15915 > 15850 = -1$  (Lebih besar dari harga *close*)

2. *Weighted Moving Average*

$15838.2 < 15850 = +1$  (Lebih kecil dari harga *close*)

3. *Momentum*

$15915 > 15850 = -1$  (Lebih besar dari harga *close*)

4. STC%K

$31.58 < 47.37 = -1$  (Lebih kecil dari nilai sebelumnya)

5. STC%D

$26.32 < 31.58 = +1$  (Lebih kecil dari nilai STC%K)

6. RSI

$48.24 < 51.07 = -1$  (Lebih kecil dari nilai sebelumnya)

7. MACD

$-167.61 = -1$  (MACD bernilai negatif)

8. *Williams' %R*

$68.42 > 52.63 = -1$  (Lebih besar dari nilai sebelumnya)

9. A/D

$-574.69 < 1600.31 = -1$  (Lebih kecil dari nilai sebelumnya)

10. CCI

$28.21 > -24.09 = +1$  (Lebih besar dari nilai sebelumnya)

11. *Alligator*

$TrendhargaTurun = -1$

Tabel 3.3 Contoh hasil diskritasi fitur data

No	Date	SMA	WMA	Momentum	STC%K	STC%D	RSI	MACD	%oR	A/D	CCI	Alligator	Direction
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	30-Sep-11	1	1	-1	1	1	-1	-1	1	-1	-1	Up	
2	3-Oct-11	1	1	-1	1	-1	-1	-1	1	-1	-1	Up	
3	4-Oct-11	1	-1	-1	-1	-1	-1	-1	1	-1	-1	Down	
4	5-Oct-11	1	1	1	-1	1	-1	1	1	1	-1	Down	
5	6-Oct-11	1	1	1	1	1	-1	1	-1	1	-1	Up	
6	7-Oct-11	1	-1	1	-1	-1	-1	-1	-1	-1	-1	Up	
7	10-Oct-11	-1	-1	1	1	-1	-1	1	-1	-1	-1	Down	
8	11-Oct-11	-1	-1	1	-1	-1	-1	1	-1	-1	-1	Down	
9	12-Oct-11	1	1	1	1	1	-1	1	1	1	-1	Down	
10	13-Oct-11	-1	1	-1	-1	1	-1	-1	-1	1	-1	Up	
...	...	...	...	...	...	...	...	...	...	...	...	...	...

### 3.4.2 Proses Learning

Proses pembuatan pohon keputusan akan dilakukan sebanyak K-iterasi yang sudah ditentukan sebelumnya. Data *training* yang digunakan untuk membuat pohon keputusan diambil sebanyak 40 hari ke belakang dari hari dimana pergerakan harga akan diprediksi. Kemudian untuk setiap pohon keputusan yang dibuat akan menggunakan subset data yang berbeda. Ukuran subset data sama besar dengan ukuran data training. Data yang berada pada subset data dipilih secara acak dari data training dan perbolehkan adanya duplikasi data. Fitur data akan dipilih secara acak sebanyak  $\sqrt{m}$  pada setiap pembagian *node*. Batas minimal jumlah data untuk melakukan pembagian data adalah sebanyak 6 data. Apabila kurang dari batas minimal maka keputusan untuk *node* tersebut akan langsung diambil berdasarkan jumlah label terbanyak.

Berikut contoh subset data yang akan digunakan untuk membuat pohon keputusan, untuk contoh akan menggunakan subset data berukuran 10 data dengan 5 fitur data. Batas minimal jumlah data untuk melakukan pembagian data pada contoh ini adalah sebanyak 4 data:

**Tabel 3.4** Contoh Data Training

No	SMA	WMA	Momentum	RSI	STC%D	Direction
1	-1	-1	-1	1	-1	Down
2	-1	-1	-1	-1	-1	Down
3	-1	1	1	1	-1	Up
4	-1	1	-1	1	1	Down
5	-1	1	-1	1	-1	Down
6	1	1	1	1	1	Up
7	1	1	1	-1	1	Up
8	1	-1	-1	-1	-1	Down
9	1	1	1	1	1	Down
10	1	1	1	1	-1	Up

Penjelasan detail proses pembuatan sebuah pohon keputusan:

### **Root Node**

Jumlah Fitur Data : 5

Fitur Data Tersisa : SMA, WMA, Momentum, RSI, STC%D

Fitur Data Dipilih Acak : RSI dan WMA

#### *Impurity Subset*

$$\text{entropy}(4,6) = -\left(\frac{4}{10}\right)\log_2\left(\frac{4}{10}\right) - \left(\frac{6}{10}\right)\log_2\left(\frac{6}{10}\right) = 0.971$$

#### *Impurity RSI*

$$\text{entropy}(3,4) = -\left(\frac{3}{7}\right)\log_2\left(\frac{3}{7}\right) - \left(\frac{4}{7}\right)\log_2\left(\frac{4}{7}\right) = 0.985$$

$$\text{entropy}(1,2) = -\left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) = 0.918$$

#### *Impurity WMA*

$$\text{entropy}(4,3) = -\left(\frac{4}{7}\right)\log_2\left(\frac{4}{7}\right) - \left(\frac{3}{7}\right)\log_2\left(\frac{3}{7}\right) = 0.985$$

$$\text{entropy}(0,3) = 0 - \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) = 0$$

#### *Information Gain*

$$\text{gain}(\text{RSI}) = 0.971 - \left(\left(\frac{7}{10}\right)*0.985 + \left(\frac{3}{10}\right)*0.918\right) = 0.061$$

$$\text{gain}(\text{WMA}) = 0.971 - \left(\left(\frac{7}{10}\right)*0.985 + \left(\frac{3}{10}\right)*0\right) = 0.2815$$

WMA memiliki nilai *information gain* terbesar, pembagian *node* dilakukan berdasarkan atribut WMA

### Level 1 (WMA = +1)

Jumlah Fitur Data : 4

Fitur Data Tersisa : SMA, Momentum, RSI, STC%D.

Fitur Data Dipilih Acak : SMA, STC%D

*Impurity Subset*

$$\text{entropy}(4,3) = -\left(\frac{4}{7}\right)\log_2\left(\frac{4}{7}\right) - \left(\frac{3}{7}\right)\log_2\left(\frac{3}{7}\right) = 0.985$$

*Impurity SMA*

$$\text{entropy}(3,1) = -\left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right) - \left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right) = 0.811$$

$$\text{entropy}(1,2) = -\left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) = 0.918$$

*Impurity STC%D*

$$\text{entropy}(2,2) = -\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) = 0.5$$

$$\text{entropy}(2,1) = 0 - \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) = 0.918$$

*Information Gain*

$$\text{gain}(\text{SMA}) = 0.985 - \left(\left(\frac{4}{7}\right)*0.811 + \left(\frac{3}{7}\right)*0.918\right) = 0.128$$

$$\text{gain}(\text{STC\%D}) = 0.985 - \left(\left(\frac{4}{7}\right)*0.5 + \left(\frac{3}{7}\right)*0.918\right) = 0.306$$

STC%D memiliki nilai *information gain* terbesar, pembagian *node* dilakukan berdasarkan atribut STC%D

### Level 1 (WMA = -1)

Jumlah Fitur Data : 4

Fitur Data Tersisa : SMA, Momentum, RSI, STC%D.

Fitur Data Dipilih Acak : SMA, STC%D

*Impurity Subset*

$$\text{entropy}(0,3) = 0 - \left(\frac{3}{3}\right)\log_2\left(\frac{3}{3}\right) = 0$$

Karena nilai *impurity* subset adalah 0, maka *node branch* ini akan menjadi *node leaf*. Keputusan diambil berdasarkan label dengan jumlah data terbanyak yaitu **Price Direction Down**.

**Level 2 (WMA = +1; STC%D = +1)**

Jumlah Fitur Data : 3

Fitur Data Tersisa : SMA, Momentum, RSI.

Fitur Data Dipilih Acak : RSI, Momentum.

*Impurity Subset*

$$\text{entropy}(2,2) = -\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) = 0.5$$

*Impurity RSI*

$$\text{entropy}(1,2) = -\left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) = 0.918$$

$$\text{entropy}(1,0) = -\left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right) - 0 = 0$$

*Impurity Momentum*

$$\text{entropy}(2,1) = -\left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) = 0.918$$

$$\text{entropy}(0,1) = 0 - \left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right) = 0$$

*Information Gain*

$$\text{gain(RSI)} = 0.5 - \left(\left(\frac{3}{4}\right)*0.918 + \left(\frac{1}{4}\right)*0\right) = -0.1885$$

$$\text{gain(Momentum)} = 0.5 - \left(\left(\frac{3}{4}\right)*0.918 + \left(\frac{1}{4}\right)*0\right) = -0.1885$$

*Momentum* memiliki nilai *information gain* terbesar, pembagian node dilakukan berdasarkan atribut *Momentum*.

**Level 2 (WMA = +1; STC%D = -1)**

Jumlah Fitur Data : 3

Fitur Data Tersisa : SMA, Momentum, RSI.

*Impurity Subset*

$$\text{entropy}(2,1) = \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) = 0.918$$

Karena jumlah data pada *node* ini tidak memenuhi batas minimal, maka *node branch* ini akan menjadi *node leaf*. Keputusan diambil berdasarkan label dengan jumlah data terbanyak yaitu **Price Direction Up**.

**Level 3 (WMA = +1; STC%D = +1; Momentum = +1)**

Jumlah Fitur Data : 2

Fitur Data Tersisa : SMA, RSI.

*Impurity Subset*

$$\text{entropy}(2,1) = \left(\frac{2}{3}\right)\log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right)\log_2\left(\frac{1}{3}\right) = 0.918$$

Karena jumlah data pada *node* ini tidak memenuhi batas minimal, maka *node branch* ini akan menjadi *node leaf*. Keputusan diambil berdasarkan label dengan jumlah data terbanyak yaitu **Price Direction Up**.

**Level 3 (WMA = +1; STC%D = +1; Momentum = -1)**

Jumlah Fitur Data : 2

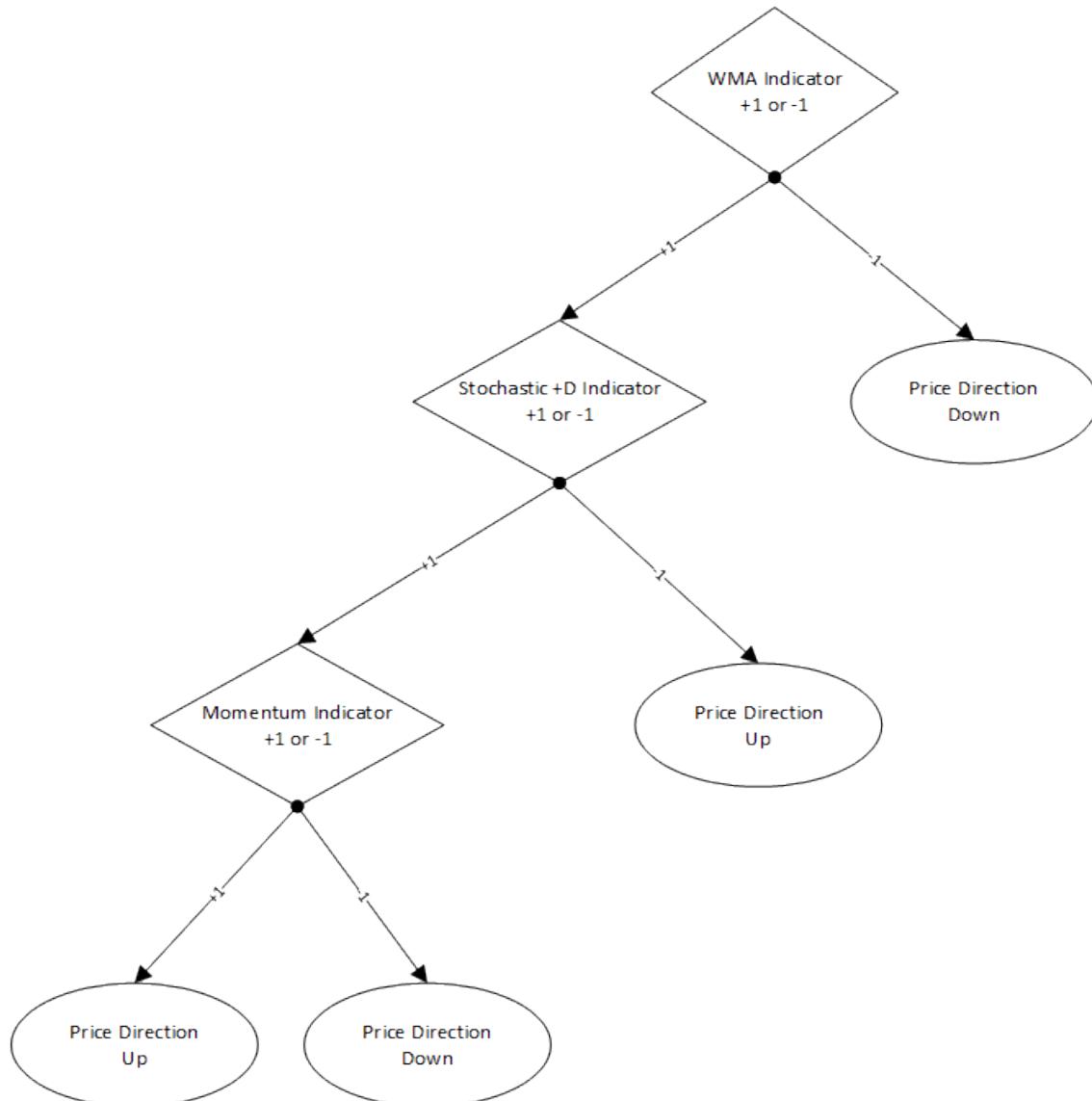
Fitur Data Tersisa : SMA, RSI.

*Impurity Subset*

$$\text{entropy}(0,1) = 0 - \left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right) = 0$$

Karena nilai *impurity* subset adalah 0, maka *node branch* ini akan menjadi *node leaf*. Keputusan diambil berdasarkan jumlah data dengan label terbanyak yaitu **Price Direction Down**.

Hasil akhir dari proses pembuatan pohon keputusan :



**Gambar 3.4** Contoh Hasil Proses *Learning* Pohon Keputusan

### 3.4.3 Klasifikasi

Pada tahap testing, setiap data testing akan diterapkan kepada semua pohon keputusan yang telah dibuat pada tahap *learning*. Contoh data yang akan diterapkan pada kumpulan pohon keputusan :

**Tabel 3.5** Contoh Data Testing

SMA	WMA	Momentum	RSI	STC%D
-1	-1	-1	1	-1

Contoh proses penerapan data kepada pohon keputusan yang telah dibuat:

1. Data masuk kepada *level root* dari pohon keputusan, cek nilai indikator WMA.
2. Nilai indikator WMA data adalah 1, turun ke *node level 1* sebelah kiri.
3. Data masuk kepada *node level 1*, cek nilai indikator STC%D.
4. Nilai indikator STC%D data adalah -1, turun ke node level 2 sebelah kanan.
5. Data masuk kepada *node level 2*, *node leaf* dengan hasil ***Price Direction Up***.

Hasil akhir proses penerapan data kepada seluruh pohon keputusan :

**Tabel 3.6** Contoh Data Testing

Pohon Keputusan	Hasil
Pohon Keputusan 1	<i>Price Direction Down</i>
Pohon Keputusan 2	<i>Price Direction Down</i>
Pohon Keputusan 3	<i>Price Direction Down</i>
Pohon Keputusan 4	<i>Price Direction Up</i>
Pohon Keputusan 5	<i>Price Direction Up</i>
Pohon Keputusan 6	<i>Price Direction Down</i>
Pohon Keputusan 7	<i>Price Direction Down</i>
Pohon Keputusan 8	<i>Price Direction Down</i>
Pohon Keputusan 9	<i>Price Direction Up</i>
Pohon Keputusan 10	<i>Price Direction Down</i>

Karena lebih banyak pohon keputusan yang memprediksi *Price Direction Down* daripada *Price Direction Up*(7 DOWN – 3 UP), maka data tersebut diprediksikan pergerakan harga saham sedang bergerak menurun(***Price Direction Down***).

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

Bab ini akan menjelaskan implementasi berdasarkan rancangan sistem, perangkat keras yang digunakan pada tahap implementasi, perangkat lunak yang diperlukan untuk menjalankan aplikasi, berserta penjelasan aplikasi.

#### **4.1 Lingkungan Implementasi**

Lingkungan implementasi sistem terbagi menjadi 2 bagian, yaitu lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

##### **4.1.1 Spesifikasi Perangkat Keras**

Deskripsi perangkat keras yang digunakan dalam tahap implementasi :

1. Notebook ASUS model G750JM
2. Prosesor Intel I7-4770HQ 4 Core 8 Threads @ 2.4Ghz
3. RAM 8GB

##### **4.1.2 Lingkungan Perangkat Lunak**

Deskripsi perangkat lunak yang digunakan dalam tahap implementasi :

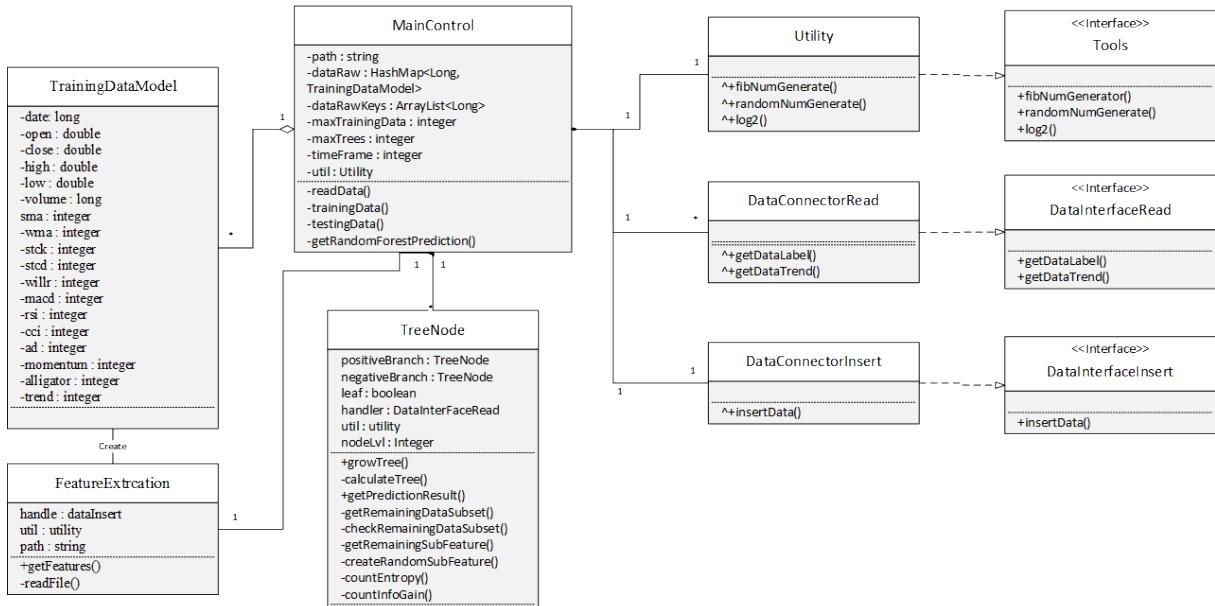
1. Sistem Operasi : Windows 8.1 64bit
2. Java SDK 1.8 64-bit
3. NetBeans IDE 8.2

#### **4.2 Implementasi Perangkat Lunak**

Bagian ini menjelaskan implementasi aplikasi prediksi trend saham untuk melakukan klasifikasi, method pemrosesan, alur pemrosesan.

#### 4.2.1 Class Diagram

Bagian ini menjelaskan tentang pemetaan *class* dan hubungan antar *class*.



Gambar 4.1 Class Diagram

#### 4.2.2 Daftar Method

Bagian ini menjelaskan tentang fungsi-fungsi yang digunakan pada setiap tahap pada aplikasi prediksi arah harga saham.

Tabel 4.1 Daftar Method

No	Nama Method	Input	Output	Keterangan
1	ReadData	-	-	Membaca dan mengekstraksi data
2	TrainingData	Integer Data Integer maxTrees	List<TreeNode>	Membangun kumpulan pohon keputusan berdasarkan data training
3	TestingData	Integer maxTrees	-	Menguji kumpulan pohon keputusan dengan mengaplikasikan data testing

**Tabel 4.1** Daftar Method

No	Nama Method	Input	Output	Keterangan
4	GrowTree	Long[] dataKey Integer[] features Boolean emptyNode	-	Membangun struktur pohon keputusan
5	CalculateTree	Long[] dataKey Integer[] features Integer featuresCount Boolean emptyNode	-	Mencari fitur data terbaik untuk menjadi pemisah node
6	CheckPurity	Long[] dataKey	Boolean	Mengecek apakah data sudah homogen atau belum
7	CreateRandom SubFeature	Integer[] features	Integer[]	Membuat subset fitur data yang dipilih secara acak
8	CountEntropy	Long[] dataKey	Double	Menghitung impurity berdasarkan entropy
9	CountInfoGain	Long[] dataKey Integer[] features	Double[]	Menghitung info gain dari setiap label data
10	GetRemaining DataSubset	Long[] dataKey Integer label	Long[]	Memisahkan data berdasarkan label fitur data yang sudah dipilih
11	CheckRemaining DataSubset	Long[] dataKey Integer label	Boolean	Mengecek sisa data berdasarkan label fitur data yang sudah dipilih
12	GetRemaining SubFeature	Integer[] features	Integer[]	Memisahkan fitur data yang sudah terpilih
13	GetFeatures	-	-	Mengekstraksi informasi indikator teknis dari data

**Tabel 4.1** Daftar Method

No	Nama Method	Input	Output	Keterangan
14	ReadFile	-	ArrayList<Tick>	Membaca data dari file berformat .csv
15	GetLabel	Integer feature Long key	Integer	Mengambil nilai label fitur data dari data
16	GetPrediction Result	Long key	Integer	Memperoleh hasil prediksi dari pohon keputusan
17	GetDataTrend	Long	Integer	Mengambil nilai trend dari data
18	InsertData	TrainingDataModel data	-	Memasukkan data yang sudah diekstrak kedalam memori
19	GetRandom ForestPrediction	TreeNode[] forest Integer lastResult	Integer	Mengklasifikasikan data berdasarkan hasil dari kumpulan pohon keputusan.

### 4.2.3 Implementasi Metode

Bagian ini menjelaskan tentang implementasi metode berdasarkan hasil perancangan dan analisis yang telah dibahas pada BabIII.

#### 4.2.3.1 Inisialisasi Data

Data yang digunakan dalam penelitian ini adalah data historis saham milik PT Gudang Garam (GGRM) dan PT Unilever Indonesia (UNVR) selama periode 2010-2015. Perusahaan-perusahaan tersebut termasuk ke dalam daftar saham LQ45, yaitu daftar 45 saham yang paling liquid dan memiliki nilai kapitalisasi yang besar di pasar saham Indonesia. Data diperoleh dari Google Finance (<https://finance.google.com/finance>).

#### 4.2.3.2 Implementasi Ekstraksi Fitur

Untuk *training* data, format data yang digunakan oleh aplikasi *adalah comma separated values*(.csv), dan memiliki header data seperti *date*, *open*, *high*, *low*, *close*, dan *volume*. Pengguna aplikasi cukup mencari lokasi data yang tersimpan di *harddisk* dan menginputkan jumlah pohon keputusan yang akan dibuat. Untuk ekstraksi informasi dari data dan pembuatan pohon keputusan akan dilakukan secara otomatis oleh aplikasi dengan bantuan *library* TA4J.

Nilai indikator teknis yang sudah didapatkan akan diubah menjadi bentuk diskrit (1,-1). Aturan yang digunakan dalam mengonversikan nilai indikator teknis sesuai dengan yang dipaparkan pada bab 2.

#### 4.2.3.3 Implementasi Proses *Learning*

Pada tahap ini, aplikasi akan membuat kumpulan pohon keputusan berdasarkan parameter yang telah ditentukan oleh pengguna aplikasi. Setiap pohon keputusan akan dibuat secara parallel dengan menggunakan fitur *multitasking* yang disediakan oleh Java untuk mengurangi waktu *training*.

Untuk menggunakan fitur *multitasking* java, aplikasi akan membuat 1 objek *Executor Service* untuk menjalankan kegiatan secara paralel. Objek *Executor Service* mempunyai parameter untuk membatasi berapa banyak kegiatan yang akan dikerjakan paralel. Pada implementasi aplikasi ini, jumlah kegiatan yang dilakukan secara paralel akan dibatasi sebanyak jumlah *core* CPU komputer yang digunakan untuk kegiatan testing. Perintah yang akan dilakukan secara parallel akan disimpan pada objek *Runnable* yang kemudian akan dijalankan oleh objek *Executor Service*.

Setiap pohon keputusan akan menggunakan *subset* data yang berbeda. Proses yang dilakukan oleh program untuk membuat sebuah pohon keputusan :

1. Mengecek kondisi berhenti. Kondisi berhenti yang di cek oleh program adalah jumlah

fitur data, jumlah data yang masih tersisa, dan *impurity* dari *subset* data.

2. Jika kondisi berhenti terpenuhi maka parameter *leaf* akan bernilai *true* dan parameter *decision* akan bernilai fitur yang dipilih. Jika tidak, lanjut ke tahap selanjutnya.
3. Memilih fitur data yang digunakan untuk membagi data secara acak
4. Menghitung *impurity* dan *infogain* setiap fitur yang terpilih dan memilih fitur yang memiliki *infogain* terbesar.
5. Setelah mendapatkan fitur data yang memiliki *infogain* terbesar, aplikasi akan membuat *subset* data sesuai dengan label fitur yang dipilih, dan kemudian akan diteruskan ke cabang selanjutnya. Proses akan diulang kembali dari awal.
6. Proses pembuatan pohon keputusan akan dilakukan hingga semua ujung cabang pohon keputusan sudah mencapai *leaf*.

### 4.2.3.4 Implementasi Klasifikasi

Pada tahap klasifikasi data terhadap metode prediksi, setiap data pada dataset testing akan diuji terhadap seluruh pohon keputusan yang akan dibuat. Hasil uji dari setiap pohon keputusan akan diakumulasikan dan dipilih berdasarkan keputusan terbanyak. Keputusan tersebut akan dimasukkan kepada *confusion matrix* untuk memperoleh hasil akurasi dari prediksi yang telah dilakukan.

## 4.3 Pengujian

Bagian ini menjelaskan tentang pengujian sistem yang telah dirancang sebelumnya. Pengujian mencakup dataset yang digunakan dan parameter yang digunakan oleh sistem.

### 4.3.1 Ketentuan Pengujian

Pengujian metode akan menggunakan 2 jenis implementasi *Random Forest* yang berbeda, yaitu tipe A dan tipe B. Perbedaan dari kedua jenis implementasi terdapat pada bagian pemilihan acak fitur data yang memilih beberapa fitur data dari m fitur data. Untuk tipe A, m adalah total fitur data yang digunakan pada *dataset* yang diuji dan nilai m tidak akan berubah ketika digunakan untuk membuat pohon keputusan. Untuk tipe B, m adalah total fitur data dari *subset* yang digunakan oleh pohon keputusan dan nilai m akan berubah sesuai dengan jumlah fitur data yang tersisa pada setiap *node* pohon keputusan.

Selama proses pengujian, parameter *Random Forest* yang digunakan adalah :

1. Banyaknya pohon keputusan yang dibuat : 10, 25, 50, 100, 200, 500, 1000.
2. Jumlah data pada subset : 40 data.

3. Aturan jumlah fitur data yang dipilih secara acak :  $\sqrt{m}$ ,  $2\sqrt{m}$ ,  $\frac{\sqrt{m}}{2}$ , dan tidak menggunakan pemilihan acak fitur data.
4. Batas minimum jumlah data setiap *node* : 6 data.
5. Jangka *Dataset* yang digunakan untuk setiap saham: 6 tahun. Data yang digunakan adalah data periode 2010 – 2015.

Parameter indikator teknis yang digunakan dalam penelitian adalah:

**Tabel 4.2** Nilai Parameter Indikator Teknis Yang Digunakan

<b>Nama Indikator Teknis</b>	<b>Parameter Yang Digunakan</b>
<i>Simple Moving Average</i>	10
<i>Weighted Moving Average</i>	10
<i>Stochastic %D</i>	3
<i>Stochastic %K</i>	10
<i>Williams' %R</i>	10
<i>Moving Average Convergence Divergence</i>	12,26
<i>Accumulation Distribution</i>	10
<i>Relative Strength Index</i>	10
<i>Community Channel Index</i>	10
<i>Momentum</i>	10
<i>Indikator Alligator</i>	5,8,13

### 4.3.2 Pengujian *Random Forest*

Untuk menguji akurasi dari sistem yang telah dirancang, penelitian ini menggunakan *f-measure* dan waktu pengujian sebagai variabel ukur. Proses pengujian akan dijalankan sebanyak 10 kali. Nilai *f-measure* dan waktu yang digunakan ketika pengujian akan di rata-ratakan. Waktu pengujian yang akan diukur adalah waktu ketika melakukan fase *training* dan fase *testing*.

#### 4.3.2.1 Pengujian Jumlah Pohon Keputusan

Pada tahap pengujian ini, parameter jumlah fitur data yang digunakan adalah  $\sqrt{m}$ . Dataset yang digunakan adalah dataset saham UNVR 2010-2015 sebanyak 1469 data dan GGRM 2010-2015 sebanyak 1469 data.

#### IV. IMPLEMENTASI DAN PENGUJIAN

---

**Tabel 4.3** Pengujian Jumlah Pohon Keputusan implementasi tipe A

No.	Data Saham	Jumlah Pohon Keputusan								<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		10	25	50	100	200	500	1000				
1	UNVR	v	-	-	-	-	-	-	87.46296	949.195	3.941	
2	UNVR	-	v	-	-	-	-	-	88.15476	1652.544	6.381	
3	UNVR	-	-	v	-	-	-	-	88.26332	2748.728	11.494	
4	UNVR	-	-	-	v	-	-	-	88.42186	5029.644	24.575	
5	UNVR	-	-	-	-	v	-	-	88.46518	9516.512	54.771	
6	UNVR	-	-	-	-	-	v	-	88.47239	23248.261	82.228	
7	UNVR	-	-	-	-	-	-	v	88.46526	46004.502	142.637	
8	GGRM	v	-	-	-	-	-	-	87.47789	994.887	4.315	
9	GGRM	-	v	-	-	-	-	-	88.16975	1660.439	7.154	
10	GGRM	-	-	v	-	-	-	-	88.34920	2791.175	12.270	
11	GGRM	-	-	-	v	-	-	-	88.48527	5064.385	23.916	
12	GGRM	-	-	-	-	v	-	-	88.46352	9717.562	53.846	
13	GGRM	-	-	-	-	-	v	-	88.48611	23630.817	107.326	
14	GGRM	-	-	-	-	-	-	v	88.49253	46817.273	190.996	

Berdasarkan hasil pengujian dengan implementasi tipe A, diperoleh nilai *f-measure* 88.47239 pada data saham UNVR dengan menggunakan 500 pohon keputusan dan 88.49253 pada data saham GGRM dengan menggunakan 1000 pohon keputusan. Waktu yang digunakan untuk memprediksi untuk proses *learning* adalah 23.2ms untuk data saham UNVR dan 46.8ms untuk data saham GGRM.

**Tabel 4.4** Pengujian Jumlah Pohon Keputusan implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan								<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		10	25	50	100	200	500	1000				
1	UNVR	v	-	-	-	-	-	-	87.45510	936.337	4.179	
2	UNVR	-	v	-	-	-	-	-	88.16938	1574.399	6.764	

**Tabel 4.4** Pengujian Jumlah Pohon Keputusan implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan							<i>F-Measure</i>	<i>Learning</i> ( $\mu\text{s}$ )	<i>Klasifikasi</i> ( $\mu\text{s}$ )
		10	25	50	100	200	500	1000			
3	UNVR	-	-	v	-	-	-	-	88.25593	2608.790	11.774
4	UNVR	-	-	-	v	-	-	-	88.42156	4732.158	24.655
5	UNVR	-	-	-	-	v	-	-	88.53716	8952.529	50.221
6	UNVR	-	-	-	-	-	v	-	88.53721	21708.784	107.569
7	UNVR	-	-	-	-	-	-	v	88.55888	42748.918	168.335
8	GGRM	v	-	-	-	-	-	-	87.49215	955.901	4.123
9	GGRM	-	v	-	-	-	-	-	88.02211	1580.279	6.648
10	GGRM	-	-	v	-	-	-	-	88.24587	2661.039	12.517
11	GGRM	-	-	-	v	-	-	-	88.54270	4806.636	27.938
12	GGRM	-	-	-	-	v	-	-	88.52841	9194.817	56.610
13	GGRM	-	-	-	-	-	v	-	88.60872	22238.164	85.388
14	GGRM	-	-	-	-	-	-	v	88.62906	43853.460	206.671

Berdasarkan hasil pengujian dengan implementasi tipe B, diperoleh nilai *f-measure* 88.55888 pada data saham UNVR dengan menggunakan 1000 pohon keputusan dan 88.62906 pada data saham GGRM dengan menggunakan 1000 pohon keputusan. Namun waktu yang digunakan untuk memprediksi untuk proses *learning* adalah 42.7ms untuk data saham UNVR dan 43.8ms untuk data saham GGRM.

Berdasarkan pengukuran *f-measure* pada implementasi tipe A, nilai f-measure akan stabil setelah 100-200 pohon keputusan. Waktu training yang digunakan untuk membuat 100-200 pohon keputusan masih dibawah 10ms. Bila dibandingkan dengan 500-1000 pohon keputusan yang membutuhkan waktu lebih dari 20ms namun tidak menghasilkan peningkatan yang signifikan. Maka 100-200 pohon dapat dipilih sebagai parameter optimal *Random Forest* implementasi tipe A.

Pada hasil pengujian tipe B, nilai *f-measure* akan semakin meningkat bila menggunakan banyak pohon keputusan. Namun peningkatan nilai *f-measure* yang hanya meningkat sebesar  $\pm 1\%$  tidak sebanding dengan lamanya waktu proses *learning* yang digunakan. Untuk hasil optimal, menggunakan 100-200 pohon keputusan pada implementasi tipe B dapat menghasilkan nilai *f-measure* yang baik dengan waktu proses *learning* yang cepat.

#### 4.3.2.2 Pengujian Jumlah Fitur Data

Pada tahap pengujian ini, parameter jumlah pohon keputusan yang digunakan adalah 100. Dataset yang digunakan adalah dataset saham UNVR 2010-2015 sebanyak 1469 data dan GGRM 2010-2015 sebanyak 1469 data.

**Tabel 4.5** Pengujian Jumlah Fitur Data implementasi tipe A

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
1	UNVR	v	-	-	-	88.42186	5029.644	24.575
2	UNVR	-	v	-	-	88.14775	7125.606	21.278
3	UNVR	-	-	v	-	88.43615	4459.179	26.362
4	UNVR	-	-	-	v	87.30362	4089.439	16.686
5	GGRM	v	-	-	-	88.48527	5064.385	23.916
6	GGRM	-	v	-	-	87.69056	7106.699	23.943
7	GGRM	-	-	v	-	88.52777	4565.550	30.685
8	GGRM	-	-	-	v	87.19518	4124.523	19.063

Berdasarkan hasil pengujian dengan implementasi tipe A, nilai *f-measure* terbesar dihasilkan sebesar 88.43615 dengan memilih acak  $\frac{\sqrt{m}}{2}$  fitur data pada data saham UNVR dan 88.52777 dengan memilih acak  $\frac{\sqrt{m}}{2}$  fitur data pada data saham GGRM.

**Tabel 4.6** Pengujian Jumlah Fitur Data implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
1	UNVR	v	-	-	-	88.42156	4732.158	24.655
2	UNVR	-	v	-	-	88.06122	6763.836	22.125
3	UNVR	-	-	v	-	88.68856	4467.029	29.454
4	UNVR	-	-	-	v	87.30362	4089.439	16.686
5	GGRM	v	-	-	-	88.54270	4806.636	27.938
6	GGRM	-	v	-	-	87.83439	6778.582	24.355

#### IV. IMPLEMENTASI DAN PENGUJIAN

---

**Tabel 4.6** Pengujian Jumlah Fitur Data implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
7	GGRM	-	-	v	-	88.54791	4520.847	30.269
8	GGRM	-	-	-	v	87.19518	4124.523	19.063

Berdasarkan hasil pengujian dengan implementasi tipe B, nilai *f-measure* terbesar dihasilkan sebesar 88.68856 dengan memilih acak  $\frac{\sqrt{m}}{2}$  fitur data pada data saham UNVR dan 88.54791 dengan memilih acak  $\frac{\sqrt{m}}{2}$  fitur data pada data saham GGRM.

Jumlah fitur data yang dipilih acak tidak terlalu berpengaruh secara signifikan pada waktu proses *learning*. Jumlah fitur data yang digunakan tidak pernah berubah. Pemilihan acak fitur data bertujuan untuk menghindari setiap pohon keputusan supaya tidak memiliki bentuk yang serupa. Mengurangi fitur yang digunakan akan dapat meningkatkan waktu training namun dengan syarat akurasi harus tidak terpengaruhi. Untuk itu, percobaan ini akan meneliti lebih lanjut fitur apa saja yang dapat dikurangi sehingga dapat meningkatkan waktu training.

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	0	10	42	40	0	0	7	0	1	0	0
1	20	43	31	14	0	0	22	1	11	0	0
2	14	16	15	22	0	0	0	15	3	0	0
3	15	7	6	7	0	0	0	21	0	0	0
4	9	4	6	4	18	0	0	1	0	0	0
5	1	0	1	2	20	18	0	0	0	0	0
6	0	0	0	0	4	20	18	0	0	0	0
7	0	0	0	0	0	4	20	18	0	0	0
8	0	0	0	0	0	0	1	0	41	0	0
9	0	0	0	0	0	0	0	0	0	42	0
10	0	0	0	0	0	0	0	0	0	0	42

**Gambar 4.2** Hasil Analisis Fitur Dataset UNVR Tanpa Pemilihan Fitur Data

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	5	8	27	16	20	0	12	4	8	0	0
1	17	35	12	27	12	2	19	6	24	19	0
2	11	19	20	13	12	2	9	10	16	7	0
3	9	13	2	10	4	3	11	10	8	4	2
4	5	5	3	4	3	2	9	14	2	2	9
5	1	4	2	5	5	5	7	10	4	5	1
6	1	3	4	1	3	12	2	6	2	5	6
7	1	1	5	3	6	4	4	2	4	5	9
8	19	6	7	2	4	3	1	2	0	0	0
9	0	0	5	6	2	11	7	1	8	3	0
10	0	0	0	0	1	3	4	3	6	8	18

**Gambar 4.3** Hasil Analisis Fitur Dataset UNVR Implementasi Tipe A

#### IV. IMPLEMENTASI DAN PENGUJIAN

---

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	5	13	26	18	16	0	10	1	8	3	0
1	19	27	20	28	21	3	14	4	19	12	0
2	13	15	12	16	13	7	11	5	10	10	0
3	8	12	5	4	2	4	10	6	4	10	5
4	5	2	2	6	9	5	5	9	9	4	3
5	3	3	2	5	5	8	3	8	5	4	3
6	2	5	2	5	4	5	2	8	1	5	5
7	3	2	4	1	1	5	2	6	6	3	8
8	1	4	4	3	2	4	4	5	4	3	5
9	7	2	5	7	4	6	5	0	0	2	0
10	0	0	0	1	1	3	4	4	4	7	14

Gambar 4.4 Hasil Analisis Fitur Dataset UNVR Implementasi Tipe B

Berdasarkan hasil analisis pada data saham UNVR, indikator alligator, momentum, SMA, RSI, dan MACD hanya berpengaruh sedikit untuk pembagian node.

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	0	0	100	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	61	0	0	0	0
2	10	0	0	0	0	0	0	0	0	19	0
3	2	3	0	0	0	0	0	0	0	0	0
4	0	2	0	3	0	0	0	0	0	0	0
5	0	0	0	2	3	0	0	0	0	0	0
6	0	0	0	0	2	3	0	0	0	0	0
7	0	0	0	0	0	2	0	3	0	0	0
8	0	0	0	0	0	0	0	2	3	0	0
9	0	0	0	0	0	0	0	0	2	3	0
10	0	0	0	0	0	0	0	0	0	0	5

Gambar 4.5 Hasil Analisis Fitur Dataset GGRM Tanpa Pemilihan Fitur Data

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	1	2	26	17	27	0	11	1	12	3	0
1	6	1	18	13	22	0	23	9	9	13	0
2	1	1	11	2	11	2	10	8	6	9	2
3	4	2	1	0	0	2	10	4	1	6	2
4	1	2	0	1	2	2	5	3	4	2	0
5	0	1	0	1	3	0	6	1	0	2	4
6	2	0	1	1	1	0	2	0	1	2	0
7	0	2	0	0	0	1	2	0	0	1	0
8	1	1	1	0	0	1	0	0	0	0	0
9	0	0	0	1	0	1	0	1	1	0	0
10	0	0	0	0	0	0	0	0	1	0	3

Gambar 4.6 Hasil Analisis Fitur Dataset GGRM Implementasi Tipe A

Level	SMA	WMA	STC%K	STC%D	Will's %R	MACD	A/D	RSI	CCI	Momentum	Alligator
0	3	6	20	12	29	0	9	5	15	1	0
1	5	3	23	8	15	1	20	16	16	11	0
2	5	5	8	6	8	1	15	11	8	9	0
3	3	4	1	3	2	1	7	1	4	8	5
4	4	1	1	5	0	1	3	4	2	7	2
5	2	6	2	1	1	4	4	1	0	1	3
6	2	0	4	1	3	4	3	1	1	2	2
7	1	1	0	3	1	4	6	3	0	0	1
8	1	0	0	0	1	2	5	1	0	2	3
9	2	1	0	3	0	0	0	0	0	2	0
10	0	0	1	0	0	2	0	1	2	1	0

Gambar 4.7 Hasil Analisis Fitur Dataset GGRM Implementasi Tipe B

Pada saham GGRM, indikator alligator, MACD, WMA, dan RSI hanya berpengaruh sedikit untuk pembagian node.

Setelah mengetahui indikator apa saja yang memiliki pengaruh paling sedikit terhadap pembentukan pohon keputusan, maka akan dilakukan tes ulang jumlah pemilihan fitur data dengan tidak menggunakan indikator yang berpengaruh sedikit terhadap pembentukan pohon keputusan.

**Tabel 4.7** Pengujian Jumlah Fitur Data implementasi tipe A

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	Klasifikasi ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
1	UNVR	v	-	-	-	88.60870	3893.377	21.079
2	UNVR	-	v	-	-	88.45034	3996.227	18.315
3	UNVR	-	-	v	-	88.75289	3910.045	21.284
4	UNVR	-	-	-	v	88.39229	3326.913	15.124
5	GGRM	v	-	-	-	88.61131	4107.722	22.406
6	GGRM	-	v	-	-	88.25303	4805.233	19.985
7	GGRM	-	-	v	-	88.57644	4084.522	22.006
8	GGRM	-	-	-	v	87.85453	3541.138	16.179

Berdasarkan hasil pengujian dengan implementasi tipe A, nilai *f-measure* terbesar dihasilkan sebesar 88.75289 dengan memilih acak  $\frac{\sqrt{m}}{2}$  fitur data pada data saham UNVR dan 88.61131 dengan memilih acak  $\sqrt{m}$  fitur data pada data saham GGRM.

**Tabel 4.8** Pengujian Jumlah Fitur Data implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	Klasifikasi ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
1	UNVR	v	-	-	-	88.58726	3919.950	20.070
2	UNVR	-	v	-	-	88.48619	4223.346	18.610
3	UNVR	-	-	v	-	88.54377	3896.539	20.154

**Tabel 4.8** Pengujian Jumlah Fitur Data implementasi tipe B

No.	Data Saham	Jumlah Pohon Keputusan				<i>F-Measure</i>	<i>Learning</i> ( $\mu S$ )	<i>Klasifikasi</i> ( $\mu S$ )
		$\sqrt{m}$	$2\sqrt{m}$	$\frac{\sqrt{m}}{2}$	All			
4	UNVR	-	-	-	v	88.39229	3326.913	15.124
5	GGRM	v	-	-	-	88.63324	4174.534	23.416
6	GGRM	-	v	-	-	88.21051	4963.247	19.420
7	GGRM	-	-	v	-	88.48096	4072.975	23.002
8	GGRM	-	-	-	v	87.85453	3541.138	16.179

Berdasarkan hasil pengujian dengan implementasi tipe B, nilai *f-measure* terbesar dihasilkan sebesar 88.58726 dengan memilih acak  $\sqrt{m}$  fitur data pada data saham UNVR dan 88.63324 dengan memilih acak  $\sqrt{m}$  fitur data pada data saham GGRM.

Pengaruh setiap indikator saham akan berbeda-beda jika digunakan pada saham yang berbeda. Pada dataset saham UNVR indikator yang digunakan adalah indikator WMA, STC%K, STC%D, Willam's%R, CCI, dan A/D. Sedangkan pada dataset saham GGRM indikator yang digunakan adalah indikator SMA, STC%K, STC%D, Willam's%R, CCI, Momentum, dan A/D. Dengan mengurangi indikator saham yang digunakan akan dapat meningkatkan hasil akhir dari prediksi arah harga saham. Berdasarkan penelitian, peningkatan waktu proses *learning* setelah mengurangi indikator saham yang digunakan  $\pm 1ms$  dengan menggunakan 100 pohon keputusan.

## **BAB V**

### **PENUTUP**

Bab ini menjelaskan kesimpulan yang diperoleh dari hasil implementasi dan pengujian sistem dan saran yang dapat menjadi dorongan untuk penelitian lebih lanjut.

#### **5.1 Kesimpulan**

Kesimpulan yang diambil dari perancangan dan implementasi sistem prediksi tren saham dengan menggunakan metode *Random Forest* adalah :

1. Semakin banyak pohon keputusan akan meningkatkan akurasi dari Random Forest namun waktu yang digunakan akan semakin lama. Berdasarkan hasil penelitian, jumlah pohon keputusan yang optimal adalah 100-200 pohon keputusan baik untuk implementasi tipe A atau implementasi tipe B.
2. Fitur data yang digunakan untuk memperoleh hasil optimal untuk setiap saham akan berbeda dan membutuhkan analisis lebih lanjut untuk mengetahui pengaruh antara indikator saham dengan data saham yang akan digunakan untuk prediksi arah harga saham. Untuk saham UNVR, indikator saham yang digunakan adalah indikator WMA, STC%K, STC%D, William's%R, CCI, dan A/D dengan parameter pemilihan acak saham  $\sqrt{m}$ . Untuk saham GGRM, indikator saham yang digunakan adalah indikator SMA, STC%K, STC%D, William's %R, A/D, CCI, dan momentum dengan parameter pemilihan acak saham  $\sqrt{m}$ .

#### **5.1.1 Saran**

Saran yang dapat dijadikan motivasi untuk melanjutkan penelitian lebih lanjut di masa mendatang adalah :

1. Menambahkan sebuah metode untuk mengukur besarnya pengaruh sebuah indikator kepada dataset saham.
2. Menambahkan indikator-indikator teknis lainnya untuk dijadikan fitur data sehingga informasi trend saham yang digunakan tidak hanya berdasarkan perspektif yang terbatas.

## DAFTAR PUSTAKA

- [1] J. G. Agrawal, V. S. Chourasia, and a K. Mittra, “State-of-the-Art in Stock Prediction Techniques,” *Adv. Res. Electr. Instrum. Eng.*, vol. 2, no. 4, pp. 1360–1366, 2013.
- [2] B. Elena-Dana and S. Ioana-Cristina, “Technical and Fundamental Anomalies. Paradoxes of Modern Stock Exchange Markets,” *Ann. Univ. Oradea, Econ. Sci. Ser.*, vol. 22, no. 1, pp. 37–43, 2013.
- [3] D. Wira, *Analisis Teknikal Untuk Profit Maksimal*. Exceed, 2010.
- [4] K. Hamid, M. T. Suleman, S. Z. Ali Shah, and R. S. Imdad Akash, “Testing the Weak Form of Efficient Market Hypothesis: Empirical Evidence from Asia-Pacific Markets,” *SSRN Electron. J.*, 2017.
- [5] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques,” *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, 2015.
- [6] C. S. Vui, G. K. Soon, C. K. On, R. Alfred, and P. Anthony, “A review of stock market prediction with Artificial neural network (ANN),” *Control System, Computing and Engineering (ICCSCE)*, 2013 IEEE International Conference on. pp. 477–482, 2013.
- [7] M. Sharma, “Survey on Stock Market Prediction and Performance Analysis,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 3, no. 1, 2014.
- [8] C. N. Babu and B. E. Reddy, “A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data,” *Appl. Soft Comput.*, vol. 23, no. October, pp. 27–38, 2014.
- [9] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, “Support vector regression with chaos-based firefly algorithm for stock market price forecasting,” *Appl. Soft Comput.*, vol. 13, no. 2, pp. 947–958, 2013.
- [10] Y. Lin, H. Guo, and J. Hu, “An SVM-based approach for stock market trend prediction,” in Proceedings of the International Joint Conference on Neural Networks, 2013.
- [11] K. A. Kumar, T. N. Babu, N. Vaishy, and K. Lavanya, “Stock Market Prediction by Non-Linear Combination based on Support Vector Machine Regression Model,” *Int. J. Adv. Res. Comput. Sci.*, vol. 7, no. 7, pp. 85–88, 2016.
- [12] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, no. july 1994. 2014.
- [13] E. Alpaydin, *Introduction to Machine Learning*, vol. 53, no. 9. 2013.

- [14] C. Sammut and G. I. Webb, Encyclopedia of Machine Learning. Boston, MA: Springer US, 2010.
- [15] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, no. 2. New York, NY: Springer New York, 2009.
- [16] S. Systems, Random Forests For Beginners. 2014.
- [17] T. M. Mitchell, Machine Learning. McGraw-Hill Education, 1997.
- [18] T. H. E. New, Finance. London: Palgrave Macmillan UK, 1989.
- [19] S. B. Achelis, Technical analysis., 1st ed. New York, NY: McGraw-Hill, 2000.

## LAMPIRAN