



ITS
Institut
Teknologi
Sepuluh Nopember

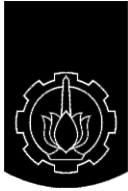
TUGAS AKHIR - EC 184801

PREDIKSI HARGA SAHAM MENGGUNAKAN MARKOV CHAIN MONTE CARLO

Alfin Alim Muhammad
NRP 07211440000021

Dosen Pembimbing
Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Mochamad Hariadi, ST., M.Sc., Ph.D.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC 184801

**PREDIKSI HARGA SAHAM MENGGUNAKAN
MARKOV CHAIN MONTE CARLO**

Alfin Alim Muhammad
NRP 0721144000021

Dosen Pembimbing
Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Mochamad Hariadi, ST., M.Sc., Ph.D.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC 184801

STOCK PRICE PREDICTION USING MARKOV CHAIN MONTE CARLO

Alfin Alim Muhammad
NRP 07211440000021

Advisor

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Mochamad Hariadi, ST., M.Sc., Ph.D.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Prediksi Harga Saham Menggunakan Markov Chain Monte Carlo**” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2020

Alfin Alim Muhammad
NRP. 0721144000021

LEMBAR PENGESAHAN

Prediksi Harga Saham Menggunakan Markov Chain Monte Carlo

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Alfin ALimMuhammad (NRP: 07211440000021)

Tanggal Ujian : 7 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh:

Dr. Supeno Mardi Susiki N., ST., MT.
NIP: 197003131995121001

(Pembimbing I)

Mochamad Hariadi, ST., M.Sc., Ph.D.
NIP: 196912091997031002

(Pembimbing II)

Dr. I Ketut Eddy Purnama, ST., MT.
NIP. 196907301995121001

(Penguji I)

Dr. Diah Puspito Wulandari, S.T., M.Sc.
NIP: 198012192005012001

(Penguji II)

Susi Juniastuti, S.T., M.Eng.
NIP: 196506181999032001

(Penguji III)



Mengetahui

Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

ABSTRAK

Nama Mahasiswa : Alfin Alim Muhammad
Judul Tugas Akhir : Prediksi Harga Saham Menggunakan
Markov Chain Monte Carlo
Pembimbing : 1. Dr. Supeno Mardi Susiki Nugroho,
ST., MT.
2. Mochamad Hariadi, ST., M.Sc., Ph.D.

Pengertian saham adalah tanda penyertaan modal seseorang atau badan usaha dalam suatu perusahaan atau perseroan terbatas. Saham menjadi investasi yang cukup menarik dan banyak dipilih penanam modal, karena memberikan tingkat keuntungan yang menarik dan cukup menantang. Ada dua analisa saham suatu perusahaan yaitu, secara fundamental dan teknikal. Analisa fundamental merupakan analisa kondisi perusahaan secara keseluruhan, baik analisa produk, keuangan, maupun pemasaran perusahaan. Sebelum memilih saham, investor harus memperhatikan analisa fundamental. Analisa teknikal dapat diartikan sebagai analisa terhadap pola pergerakan harga di masa lampau dengan tujuan untuk meramalkan harga di masa yang akan datang. Pada prediksi ini menggunakan analisa saham secara technical menggunakan simulasi *Markov Chain Monte Carlo* (MCMC) yang diaplikasikan pada *Apache Spark*. Dengan metode ini, MCMC akan menganalisa kemungkinan pola-pola apa saja yang mungkin terjadi di masa mendatang berdasarkan pola-pola yang pernah terjadi sebelumnya.

Kata Kunci : Harga Saham, Analisa Teknikal, MCMC (*Markov Chain Monte Carlo*), *Apache Spark*

ABSTRACT

Name : Alfin Alim Muhammad
Title : *Stock Price Prediction Using Markov Chain Monte Carlo*
Advisors : 1. Dr. Supeno Mardi Susiki Nugroho, ST., MT.
2. Mochamad Hariadi, ST., M.Sc., Ph.D.

Definition of shares is a sign of capital participation of a person or business entity in a company or limited liability company. Shares become an attractive investment and many investors choose it, because it provides an attractive and quite challenging level of profit. There are two stock analysis of a company that is, fundamentally and technically. Fundamental analysis is an analysis of the overall condition of the company, both product analysis, financial, and corporate marketing. Before choosing stocks, investors must pay attention to fundamental analysis. Technical analysis can be interpreted as an analysis of patterns of price movements in the past with the aim to predict prices in the future. In this prediction using technical stock analysis using Markov Chain Monte Carlo simulation (MCMC) which is applied to Apache Spark. With this method, MCMC will analyze the possible patterns of what might happen in the future based on patterns that have happened before.

Keywords : Stock, Technical, MCMC, Apache Spark.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Prediksi Harga Saham Menggunakan Markov Chain Monte Carlo**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. dan Bapak Mochamad Hariadi, ST., M.Sc., Ph.D. atas bimbingan selama mengerjakan penelitian.
3. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
4. Seluruh teman-teman *B201-crew* Laboratorium Bidang Studi Teknik Komputer dan Telematika.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak	vi
Abstract	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	3
1.5 Sistematika Penulisan	3
1.6 Relevansi	4
2 TINJAUAN PUSTAKA	5
2.1 Data	5
2.1.1 Data Terstruktur	5
2.1.2 Data Tidak Terstruktur	5
2.2 Apache Spark	6
2.2.1 Arsitektur Spark	6
2.2.2 Spark Library	7
2.2.3 Cluster Mode	8
2.3 Saham	10
2.3.1 Jenis Saham	11
2.3.2 Analisa Teknikal Saham	13
2.4 Bayesian Inference	14
2.5 Geometric Brownian Motion	15
2.6 Student-T Distribution	15
2.7 Markov Chain	16
2.8 Monte Carlo	18

2.9	Markov Chain Monte Carlo	18
2.9.1	The Metropolis-Hastings Algorithm	19
2.10	PyMC3 Python	19
3	DESAIN DAN IMPLEMENTASI SISTEM	21
3.1	Desain Sistem	21
3.2	Data yang Digunakan	24
3.3	Pembuatan Apache Spark Multinode Cluster	25
3.3.1	Desain Jaringan	26
3.4	Perancangan Metodologi Prediksi	28
3.4.1	Akuisisi Data	28
3.4.2	Pemodelan Data Saham	31
3.4.3	Model Fitting	34
3.4.4	Posterior Predictive Check	36
4	PENGUJIAN DAN ANALISA	39
4.1	Evaluasi Model	39
4.1.1	Pengujian Model Student-T distribution	39
4.2	Pengujian Waktu Komputasi pada Single Computer	41
5	PENUTUP	45
5.1	Kesimpulan	45
5.2	Saran	45
	DAFTAR PUSTAKA	47
	Biografi Penulis	49

DAFTAR GAMBAR

2.1	Arsitektur dasar <i>Apache Spark</i>	6
2.2	<i>Library</i> pada <i>Apache Spark</i>	7
2.3	Arsitektur <i>Apache Spark Standalone Cluster</i>	8
2.4	Arsitektur <i>Hadoop YARN Cluster</i>	9
2.5	Arsitektur <i>Apache Mesos Cluster</i>	10
2.6	Contoh data harga saham pada pasar sekunder . . .	10
2.7	Pergerakan harga saham BCA pada bulan Mei-September 2019	11
2.8	Grafik harga saham BCA bentuk <i>Candlestick</i> . . .	13
3.1	Alur kerja penelitian	21
3.2	Rancangan sistem layanan yang digunakan	23
3.3	Data saham Bank BCA tahun 2004-2019	24
3.4	Pergerakan harga saham	25
3.5	Desain jaringan	26
3.6	Tampilan <i>web</i> untuk memantau <i>Apache Spark</i>	29
3.7	Alur implementasi prediksi	30
3.8	<i>Web page of Yahoo Finance</i>	31
3.9	Pengambilan data saham Bank BCA dari <i>yahoo finance</i>	31
3.10	Nilai <i>daily return</i> saham Bank BCA	32
3.11	Grafik <i>daily return</i> saham Bank BCA	32
3.12	Histogram <i>daily return</i> saham Bank BCA	33
3.13	<i>Data model</i>	34
3.14	<i>Data tuning</i>	35
3.15	<i>Tracing</i>	35
3.16	<i>Tracing Summary</i>	36
3.17	<i>Posterior predictive check</i>	36
3.18	<i>Histogram of posterior predictive check</i>	37
4.1	Histogram data asli	40
4.2	Histogram data prediksi	40
4.3	Waktu komputasi menggunakan <i>NUTs sampler</i> . . .	41
4.4	Waktu komputasi menggunakan <i>Metropolis sampler</i>	44

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	Alokasi <i>ip address</i> tiap <i>node</i>	27
4.1	Hasil pengujian berdasarkan jumlah data	41
4.2	Harga aktual saham GSPC bulan Januari 2016 ber- sumber dari <i>finance.yahoo.com</i>	42
4.3	Spesifikasi perangkat pengujian pada laboratorium Telematika B201 Teknik Komputer ITS	43
4.4	Spesifikasi perangkat pengujian menggunakan laptop pribadi	43
4.5	Spesifikasi perangkat pengujian menggunakan layan- an Cloud Service	43
4.6	Waktu pengujian pada <i>Apache Spark</i>	43

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Saham merupakan tanda bukti penyertaan modal dalam suatu perusahaan atau perseroan. Jadi ketika kita membeli saham suatu perusahaan sama saja kita memiliki perusahaan tersebut dan klaim atas pendapatan serta asset perusahaan tersebut. Perdagangan saham terjadi di Bursa Efek Indonesia yaitu tempat berlangsungnya perdagangan efek di pasar sekunder. Untuk dapat melakukan transaksi, *investor* harus menjadi nasabah di perusahaan efek tertentu. Setelah menjadi nasabah di perusahaan efek, investor bisa memilih saham perusahaan apa yang akan dibeli oleh *investor*.

Ada 2 macam analisa yang biasa dipakai dalam perdagangan saham yaitu analisa fundamental dan analisa teknikal. Analisa fundamental adalah analisa mengenai kondisi perusahaan yang mencakup pendapatan, pengeluaran, produk, serta strategi pemasaran perusahaan. Analisa teknikal adalah salah satu analisis atau metode pendekatan yang mengevaluasi pergerakan suatu harga saham, valas, kontrak berjangka, indeks dan instrumen keuangan lainnya [1] Analisa fundamental digunakan oleh investor untuk memilih perusahaan mana yang sahamnya akan dibeli. Analisa teknikal digunakan untuk meramalkan harga di masa yang akan datang berdasarkan pola pergerakan harga di masa lampau. Analisa teknikal diipakai oleh investor untuk menentukan kapan harus membeli dan kapan harus menjual sahamnya.

Markov Chain Monte Carlo (MCMC) merupakan metode untuk meramalkan probabilitas terjadinya suatu peristiwa di masa yang akan datang berdasarkan peristiwa saat ini atau peristiwa-peristiwa yang pernah terjadi sebelumnya. dengan menghitung probabilitas peristiwa-peristiwa yang mungkin terjadi lalu disimulasikan menggunakan metode MCMC, maka akan didapatkan peristiwa

mana yang sering muncul selama simulasi, peristiwa itulah yang akan menjadi hasil peramalan. Metode MCMC ini membutuhkan banyak prosedur komputasi atau perhitungan sehingga jika menggunakan komputer biasa akan membutuhkan waktu yang cukup lama. Sehingga butuh komputer dengan kemampuan komputasi yang cepat.

Saat ini teknologi komputasi *big data* dilakukan secara terdistribusi dengan bantuan engine komputasi terdistribusi yang menggabungkan beberapa komputer untuk melakukan tugas pengolahan data. Salah satu contoh engine komputasi terdistribusi adalah *Apache Spark*. *Apache Spark* memiliki kemampuan yang tidak dimiliki oleh engine-engine lainnya yaitu untuk mengolah data secara realtime. *Apache Spark* juga memiliki kecepatan yang tinggi dibandingkan dengan engine lainnya serta mendukung komputasi untuk kecerdasan buatan atau *Artificial Intelligence*.

1.2 Permasalahan

Setiap saham memiliki pola pergerakan harga yang kadang sulit diprediksi pergerakannya di masa yang akan datang. Oleh karena itu, diperlukan metode yang mampu mengenali pola pergerakan harga di masa lampau untuk meramalkan pergerakan harga di masa yang akan datang.

Untuk mengidentifikasi pola pergerakan harga saham membutuhkan komputasi dan perhitungan yang kompleks, sehingga membutuhkan sumber daya komputasi yang handal. Komputasi yang kompleks membutuhkan komputer dengan spesifikasi yang tinggi dan mahal.

1.3 Tujuan

Tujuan penelitian tugas akhir ini dilakukan untuk mengimplementasikan Apache Spark dalam memprediksi harga saham menggunakan metode *Markov Chain Monte Carlo* (MCMC) serta menguji performa Apache Spark dalam melakukan komputasi. Dengan menggunakan *Apache Spark* komputasi dilakukan dengan membagi beban kerja pada beberapa komputer atau yang lebih dikenal dengan komputasi paralel. Dengan penelitian ini diharapkan dapat mempercepat proses komputasi serta membantu *investor* da-

lam mengambil keputusan ketika menanamkan modalnya di suatu perusahaan.

1.4 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah :

1. Program berbasis *Apache Spark* akan dijalankan dengan menggabungkan *resources* beberapa komputer
2. Pengolahan data tidak terjadi secara realtime

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

2. BAB II Dasar Teori

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait teknologi komputasi awan, mesin virtual, dan teori-teori penunjang lainnya.

3. BAB III Perancangan Sistem dan Impementasi

Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implentasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisa sistem. Spesifikasi perangkat keras dan perangkat lunak yang diuji juga disebutkan dalam bab ini. Sehingga ketika akan dikem-

bangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi dengan mudah tanpa harus melakukan ujicoba perangkat lunak maupun perangkat keras lagi.

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai prediksi harga saham merupakan bidang penelitian yang jarang dilakukan. Aplikasi yang biasa digunakan hanya menampilkan grafik harga harian saham, sehingga membutuhkan analisa lebih lanjut lagi. Dari penelitian ini dihasilkan sebuah program yang mampu menganalisa probabilitas harga naikturunya harga saham bagi *investor*. Tujuannya diperoleh referensi bagi pengguna dalam pemilihan saham yang menguntungkan.

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Data

Data adalah sejumlah fakta yang dikumpulkan untuk tujuan tertentu. Data dapat berupa teks dan angka, gambar, suara, atau video. Data yang tidak diolah tidak akan berguna karena kecenderungan data yang disimpan tidak mempunyai struktur dan relasi antara satu sama lain [2], proses di mana data dikumpulkan, ditransmisikan, dan disimpan beragam, metode penyimpanan yang beragam, dan tipe data yang beragam, sehingga harus diolah sedemikian rupa sehingga orang dapat menarik suatu kesimpulan berdasarkan data tersebut. Data digolongkan menjadi dua, yaitu:

2.1.1 Data Terstruktur

Data terstruktur adalah data yang bergantung pada model data dan menempati *field* yang tetap dalam sebuah *record*. Oleh karena itu, data terstruktur biasanya disimpan dalam *database* atau *spreadsheet*. Data terstruktur perlu dibuat model data terlebih dahulu, sebuah model dari jenis data yang akan disimpan dan bagaimana data tersebut disimpan, diproses, dan diakses. Dalam hal ini termasuk mendefinisikan *field* dari data, tipe data (numerik, alfabet, tanggal, dan lain-lain) dan batasan lain (misalnya karakter yang diperbolehkan, batas angka maksimum, dan lain-lain).

2.1.2 Data Tidak Terstruktur

Data tidak terstruktur adalah data yang tidak memiliki model data[3]. Data tipe ini biasanya banyak mengandung teks, dapat juga mengandung informasi lain seperti tanggal, angka, konten multimedia dan lain sebagainya sehingga menyebabkan ketidakteraturan dan ambiguitas pada *dataset*. Contoh dari data tidak terstruktur antara lain, dokumen *word processor*, video, foto, file audio, file presentasi, laman web dan berbagai jenis dokumen bisnis.

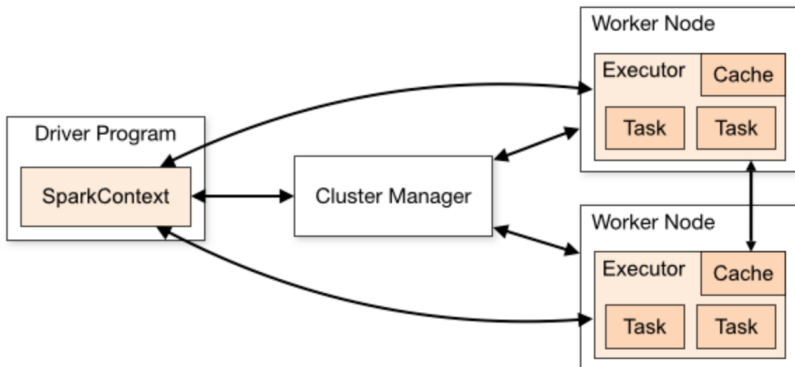
2.2 Apache Spark

Apache Spark merupakan perangkat lunak yang secara khusus digunakan untuk mengolah data dalam skala besar hingga ratusan *gigabytes* dengan teknologi komputasi kluster terdistribusi. Dengan *Apache Spark* kita dapat menggabungkan beberapa komputer untuk mengolah data. *Apache Spark* mampu mengolah data 100 kali lebih cepat dari *Hadoop*, karena eksekusi berada dalam memori sehingga mengurangi perpindahan data pada *disk*. *Apache Spark* memiliki dua abstraksi utama yaitu :

1. *Resilient Distributed Dataset* (RDD) merupakan elemen data yang tidak dapat diubah namun dapat diduplikasi ke bentuk lain. RDD dapat dibagi menjadi beberapa bagian lalu dieksekusi pada beberapa node pekerja yang berbeda.
2. *Directed Acyclic Graph* (DAG) merupakan bagian penjadwalan. Dibandingkan dengan MapReduce yang membuat grafik dalam dua tahap *Map and reduce*, *Apache Spark* dapat membuat DAG yang memiliki banyak tahapan.

2.2.1 Arsitektur Spark

Arsitektur dasar *Apache Spark* dapat dilihat pada Gambar 2.1.

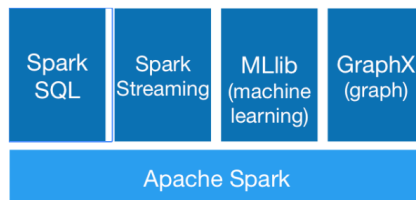


Gambar 2.1: Arsitektur dasar *Apache Spark*

1. *Driver Program* berfungsi memanggil program utama *Apache Spark* lalu membuat *SparkContext*. Driver program juga berisi komponen lain seperti DAG dan manajer blok yang menerjemahkan kode program dari pengguna ke dalam bahasa mesin yang dieksekusi di node pekerja.
2. *SparkContext* merupakan *gateway* untuk semua fungsi Spark. Semua perintah yang berjalan di Spark akan melewati ini. Driver Program dan *SparkContext* secara bersama-sama mengawasi pelaksanaan pekerjaan di dalam kluster
3. *Cluster Manager* merupakan pembagi tugas dan mendistribusikannya pada node pekerja.
4. *Worker Node* merupakan node pekerja yang mengeksekusi perintah dari *SparkContext*. Eksekusi ini terjadi pada RDD yang dipartisi lalu hasilnya dikembalikan pada *SparkContext*.

2.2.2 Spark Library

Dalam *Apache Spark* terdapat beberapa *library* dasar yang membantu kita dalam mengolah data. *Library* tersebut dapat dilihat pada Gambar 2.2.



Gambar 2.2: *Library* pada *Apache Spark*

1. *SparkSQL* merupakan *library* yang menyediakan abstraksi data terstruktur maupun data tidak terstruktur. *Library* ini juga mampu mengeksekusi SQL *queries*. *Spark SQL* juga mampu membaca data dari *Apache HIVE*. *SparkSQL* mengeksekusi data lebih cepat daripada HIVE karena proses berjalan di memori sehingga mengurangi proses masuk dan keluarnya data di dalam *disk*. *SparkSQL* memiliki 3 layer komponen yaitu

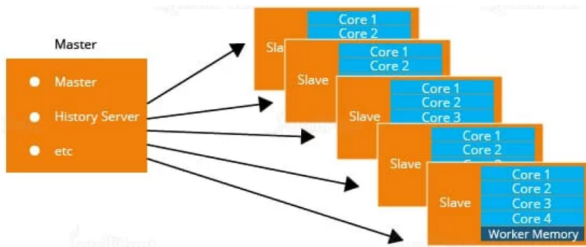
- (a) *Language API*
 - (b) *Schema RDD*
 - (c) *Data Sources*
2. *Spark Streaming* merupakan komponen yang menyediakan pemrosesan data secara *realtime*. Melalui API aliran data dimanipulasi lalu diproses untuk menjadi sebuah hasil analisa secara langsung.
 3. *MLlib* menyediakan pustaka untuk algoritma kecerdasan buatan, klasifikasi, penyaringan data dan sebagainya.
 4. *GraphX* menyediakan fitur API untuk *graph* dan *graph-parallel computation*.

2.2.3 Cluster Mode

SparkContext pada *Apache Spark* dapat bekerja dengan beberapa *cluster manager* seperti *Standalone Cluster Manager*, *Yet Another Resource Navigator (YARN)* atau *Mesos*[3].

1. *Standalone Cluster*

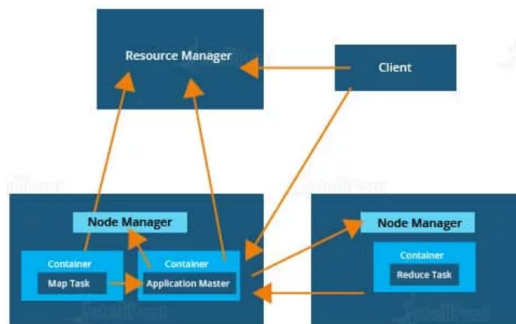
Pada *Standalone Cluster* yang menjadi *Resources Manager* adalah *Standalone Master* yang mengatur semua sumber daya pada tiap node pekerja. Ketika sebuah program dijalankan akan melakukan koneksi ke *Standalone Master* untuk menyediakan sumber daya yang dibutuhkan untuk menjalankan sebuah program. Dengan *mode* ini tersedia juga *Web UI* untuk melihat semua node dan status tiap pekerjaan pada tiap node. Arsitektur *Standalone Master* dapat dilihat pada Gambar 2.3.



Gambar 2.3: Arsitektur *Apache Spark Standalone Cluster*

2. Hadoop YARN

Pada *Hadoop YARN* ada dua komponen penting yang menjadi *Resources Management*. *Resources Manager* mengatur semua sumber daya pada semua aplikasi, serta mengatur pedjadwalan pada sebuah aplikasi yang berjalan. *Node Manager* terdiri dari dua komponen *Application Manager* dan *Container*. Setiap tugas berjalan pada *Container*. Setiap pekerjaan atau tugas membutuhkan satu atau beberapa *Container*. *Node Manager* akan memantau semua pekerjaan pada *Container* dan sumber daya yang digunakan lalu dilaporkan pada *Resources Manager*. *YARN* juga menyediakan keamanan untuk autentikasi pada *web console* dengan menggunakan *Kerberos*. Arsitektur *Hadoop YARN* dapat dilihat pada Gambar 2.4.

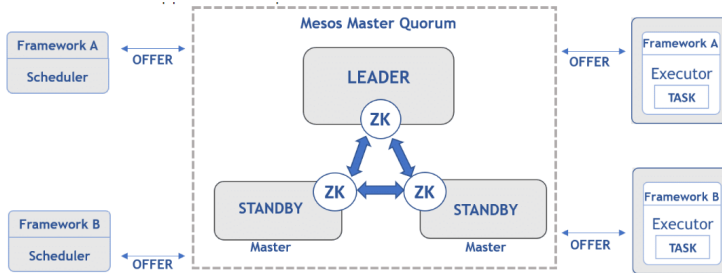


Gambar 2.4: Arsitektur *Hadoop YARN Cluster*

3. Apache Mesos

Apache Mesos menangani semua beban kerja dari berbagai node dengan *resource sharing and isolation*. Dengan demikian membantu mengatur aplikasi pada skala besar yang membutuhkan sumber daya yang sangat banyak. *Apache Mesos* memiliki tiga komponen yaitu *Mesos Master*, *Mesos Slave*, *Mesos Framework*. *Mesos Master* menyediakan toleransi kegagalan dan *recovery loss* setelah terjadi kegagalan yang mungkin disebabkan oleh listrik padam. Pada jenis kluster ini akan terdapat banyak *Mesos Master*. *Mesos Slave* merupakan sebuah

unit yang menyediakan sumber daya pada *Mesos Master* hanya jika *Mesos Master* memberi sebuah tugas atau pekerjaan. *Mesos FrameWorks* memperbolehkan sebuah aplikasi untuk meminta sumber daya dari *cluster* untuk menjalankan sebuah tugas atau pekerjaan. Arsitektur *Mesos Cluster* dapat dilihat pada Gambar 2.5.



Gambar 2.5: Arsitektur *Apache Mesos Cluster*

2.3 Saham

Date	Open	High	Low	Close*	Adj Close**	Volume
Nov 22, 2019	31,400.00	31,550.00	31,350.00	31,525.00	31,525.00	11,207,000
Nov 21, 2019	31,750.00	31,750.00	31,500.00	31,500.00	31,500.00	11,504,700
Nov 20, 2019	31,275.00	31,750.00	31,275.00	31,750.00	31,750.00	8,645,900
Nov 19, 2019	31,650.00	31,750.00	31,500.00	31,575.00	31,575.00	12,023,000
Nov 18, 2019	-	-	-	-	-	-
Nov 15, 2019	31,400.00	31,450.00	31,275.00	31,375.00	31,375.00	9,427,600
Nov 14, 2019	31,325.00	31,400.00	31,050.00	31,350.00	31,350.00	7,572,400
Nov 13, 2019	31,325.00	31,425.00	31,325.00	31,400.00	31,400.00	8,016,700
Nov 12, 2019	31,100.00	31,400.00	31,100.00	31,325.00	31,325.00	9,527,100

*Close price adjusted for splits. **Adjusted close price adjusted for both dividends and splits.

Gambar 2.6: Contoh data harga saham pada pasar sekunder

Saham dapat didefinisikan sebagai tanda penyertaan modal seseorang atau pihak (badan usaha) dalam suatu perusahaan atau perseroan terbatas[1]. menerbitkan saham merupakan

salah satu pilihan perusahaan dalam melakukan pendanaan perusahaan. Keuntungan saham dapat diperoleh dari dividen dan *capital gain*. Dividen merupakan pembagian keuntungan dari perusahaan. Untuk mendapatkan dividen seseorang harus memegang saham dalam waktu yang relatif lama hingga kepemilikan saham tersebut diakui untuk mendapatkan dividen dalam Rapat Umum Pemegang Saham (RUPS). *Capital Gain* merupakan selisih antara harga beli dan harga jual akibat adanya aktivitas perdagangan saham di pasar sekunder.



Gambar 2.7: Pergerakan harga saham BBCA pada bulan Mei-September 2019

2.3.1 Jenis Saham

Saham dapat dikelompokkan dalam beberapa jenis yaitu :

1. Berdasarkan hak tagih
 - (a) Saham biasa (*Common Stock*)
Saham ini merupakan saham yang diperjual-belian di pasar modal. Pemegang saham memiliki kewajiban terbatas. Apabila perusahaan bangkrut kerugian maksimum ditanggung oleh pemilik saham sebesar modal yang ditanamkan.
 - (b) Saham preferen (*Preferred Stock*)
Merupakan saham gabungan antara obligasi dan saham

biasa. Dapat menghasilkan pendapatan tetap. Jika perusahaan bangkrut modal yang ditanamkan pemegang saham akan dikembalikan terlebih dahulu.

2. Berdasarkan cara pengalihannya

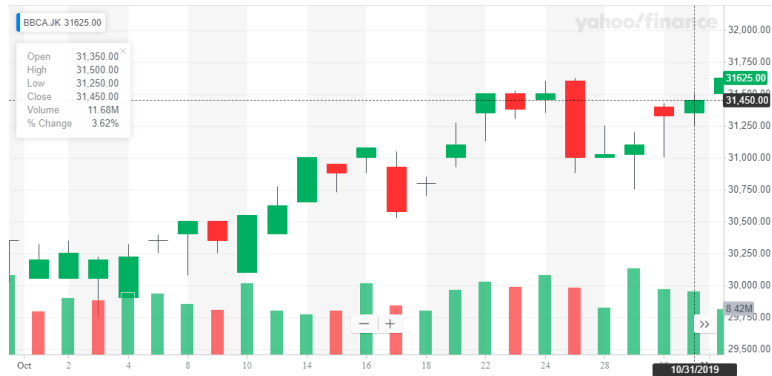
- (a) Saham atas unjuk
Yaitu saham yang tidak tertulis nama pemiliknya, agar mudah dijual-belikan di pasar modal.
- (b) Saham atas nama
Saham yang ditulis nama pemiliknya, untuk memindahkannya harus melalui prosedur tertentu.

3. Berdasarkan kinerja perdagangan

- (a) *Blue-Chips Stock*
Merupakan saham dari suatu perusahaan yang mempunyai reputasi tinggi atau sebagai pemimpin di industri sejenis, memiliki pendapatan stabil dan konsisten.
- (b) *Income Stock*
Saham yang memiliki kemampuan membayar dividen yang lebih tinggi dari rata-rata dividen yang dibayarkan tahun lalu.
- (c) *Growth Stock*
Merupakan saham yang berasal dari daerah tertentu yang kurang populer di kalangan emiten.
- (d) *Speculative Stock*
Merupakan saham yang tidak bisa konsisten memperoleh penghasilan dari tahun ke tahun, akan tetapi memiliki penghasilan yang tinggi di masa mendatang
- (e) *Counter Cyclical Stock*
Merupakan saham yang tidak terpengaruh kondisi ekonomi makro maupun situasi bisnis secara umum. Harga saham ini tetap tinggi di mana emitennya mampu memberikan dividen yang tinggi.

2.3.2 Analisa Teknikal Saham

Analisa Teknikal merupakan suatu teknik analisa yang dikenal dalam bidang keuangan untuk memprediksi *trend* suatu harga saham dengan cara mempelajari data pasar yang lampau terutama pada pergerakan dan volume. Analisa teknikal dapat menggunakan berbagai model, misalnya Indeks Kekatan Relatif untuk analisa pergerakan harga, Indeks Pergerakan Ratarata, regresi untuk menganalisa pola grafik. Analisa teknikal dikenal secara luas oleh para pedagang saham atau yang biasanya disebut *trader*. Untuk menggunakan analisa teknikal maka diperlukan sebuah data visual harga saham yang biasanya disajikan dalam bentuk grafik. Grafik yang digunakan biasanya menggunakan *candlestick* seperti Gambar 2.8.



Gambar 2.8: Grafik harga saham BBKA bentuk Candlestick

Pada gambar 2.8 warna hijau menunjukkan kenaikan harga, puncak garis vertikal pada warna hijau menunjukkan harga tertinggi saat *trading*, sedangkan pada bagian bawah menunjukkan harga terendah saat *trading*, titik pada garis vertikal yang berpotongan dengan warna hijau pada bagian atas adalah harga penutupan saham saat *trading* ditutup dan pada bagian bawah menunjukkan harga pembukaan saham saat *trading* pertama dibuka.

Warna merah menunjukkan penurunan harga, puncak garis vertikal pada warna merah menunjukkan harga tertinggi saat *trading*, sedangkan pada bagian bawah menunjukkan harga terendah

saat *trading*, titik pada garis vertikal yang berpotongan dengan warna merah pada bagian atas adalah harga pembukaan saham saat *trading* pertama dibuka dan pada bagian bawah menunjukkan harga penutupan saham saat *trading* ditutup.

2.4 Bayesian Inference

Teorema Bayes merupakan sebuah persamaan matematika untuk menghitung probabilitas terjadinya suatu kejadian berdasarkan hasil observasi sebelumnya. Nama Teorema Bayes diambil dari nama penemu teori tersebut yaitu Reverend Thomas Bayes (1702-1761)[4]. Teorema Bayes banyak diaplikasikan pada bidang komputasi cerdas sebagai dasar metode *machine learning* dan *data mining*.

Bayesian Inference merupakan metode statistika yang menggunakan teorema Bayes untuk memperbarui hipotesa sebagai bukti dari yang observasi. *Bayesian Inference* banyak digunakan pada berbagai bidang. Pada pengambilan keputusan *Bayesian Inference* merujuk pada probabilitas subjektif atau *Bayesian Probability*.

Jika $P(\theta)$ menyatakan probabilitas kejadian A, $P(y)$ menyatakan probabilitas kejadian B, dan probabilitas kejadian B terjadi jika kejadian A telah terjadi adalah $P(y|\theta)$ maka persamaannya adalah :

$$P(y|\theta) = \frac{P(\theta \cap y)}{P(y)}$$

$$\Leftrightarrow P(\theta \cap y) = P(\theta|y)P(y)$$

Karena sifat komutatif

$$P(\theta \cap y) = P(y \cap \theta)$$

sehingga

$$P(\theta|y)P(y) = P(y|\theta)P(\theta)$$

lalu bentuk persamaannya menjadi seperti persamaan 2.1

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)} \tag{2.1}$$

dengan menghilangkan $P(y)$ maka

$$P(\theta|y) \propto P(y|\theta)P(\theta) \quad (2.2)$$

Distribusi prediksi posterior merupakan distribusi data baru yang dibatasi oleh posterior

$$P(x|X, \alpha) = \int P(x|\theta) P(\theta|X, \alpha) d\theta \quad (2.3)$$

Distribusi prediksi prior merupakan distribusi data baru yang dibatasi oleh posterior

$$P(x|\alpha) = \int P(x|\theta) P(\theta|\alpha) d\theta \quad (2.4)$$

Dimana x merupakan titik data yang akan diprediksi, X merupakan data observasi, α merupakan hiperparameter dari parameter distribusi, dan θ merupakan parameter data titik distribusi data *daily return* harga saham.

2.5 Geometric Brownian Motion

Geometric Brownian Motion adalah gerakan terus-menerus dari partikel-partikel benda padat, cair maupun gas. *Brownian Motion* pertama kali dibuktikan oleh Robert Brown seorang botanis skotlandia pada tahun 1827 [5]. Brown mengamati gerakan partikel-partikel ini dengan mikroskop, lalu ia juga menemukan bahwa pergerakan partikel-partikel ini akan semakin cepat apabila suhu bertambah. Portofolio kumulatif *return* dari harga saham dimodelkan dengan *Geometric Brownian Motion*. Persamaan *Geometric Brownian Motion* merupakan persamaan turunan dari proses stokastik seperti pada Persamaan 3.1.

2.6 Student-T Distribution

Distribusi *Student-T* atau distribusi T merupakan distribusi variabel acak kontinu selain distribusi normal[6]. Distribusi probabilitas *Student-T* diturunkan dari distribusi probabilitas normal

baku. Fungsi probabilitas kepadatan *Student-T* adalah :

$$f(x|\mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu}\right)^{\frac{1}{2}} \left[1 + \frac{\lambda(x - \mu)^2}{\nu}\right]^{-\frac{\nu+1}{2}} \quad (2.5)$$

dengan ν adalah derajat kebebasan, μ parameter lokasi, λ parameter skala. Apabila λ tidak diketahui dapat diganti dengan σ standar deviasi.

2.7 Markov Chain

Model *Markov Chain* dikembangkan oleh A.A Markov pada 1896 [?]. Dalam analisa model *markov* dihasilkan suatu informasi probabilitas pada waktu tertentu yang membantu untuk mengambil keputusan. Apabila diketahui suatu kejadian waktu sekarang maka peluang kejadian setelahnya bergantung pada kejadian sekaraang.

$$K_{t(j)} = K_{t(j-1)}xP \quad (2.6)$$

$K_{t(j)}$: Peluang kejadian pada $t(j)$

P : Probabilitas transisional

$t(j)$: Waktu ke- j

Syarat-syarat dalam analisa markov adalah :

1. Jumlah probabilitas transisi untuk suatu keadaan awal dari sistem sama dengan 1.
2. Probabilitas-probabilitas tersebut berlaku untuk semua partisipan dalam sistem.
3. Probabilitas transisi konstan sepanjang waktu.
4. Kondisi merupakan kondisi yang independen sepanjang waktu.

Contoh kasus, sebuah survei dilakukan di sebuah wilayah di kota Jakarta. Diketahui bahwa wilayah tersebut terdiri dari 1000 keluarga. Dari survei tersebut, diperoleh data bahwa 600 keluarga

merupakan pelanggan toserba ‘Serba’ dan 400 keluarga merupakan pelanggan toserba ‘Ada’. Pada bulan itu, diketahui bahwa dari 600 keluarga pelanggan toserba ‘Serba’ diperoleh data bahwa 400 keluarga tetap berbelanja di toserba ‘Serba’ dan 200 lainnya berbelanja di toserba ‘Ada’. Dari 400 keluarga pelanggan toserba ‘Ada’ dinyatakan bahwa 150 keluarga tetap berbelanja di toserba ‘Ada’. Sedangkan 250 lainnya berbelanja di toserba ‘Serba’.

Hitunglah Probabilitas untuk toko “Serba” dan “Ada” pada bulan ketiga apabila pada bulan pertama keluarga tersebut memilih untuk berbelanja di toko “Serba”. Probabilitas untuk toko “Serba” dan “Ada” pada bulan ketiga apabila pada bulan pertama keluarga tersebut memilih untuk berbelanja di toko “Ada”.

Jawaban :

Langkah pertama yang perlu dilakukan untuk menyelesaikan seluruh pertanyaan di atas adalah dengan menentukan matriks transisi untuk menghitung nilai probabilitas.

Probabilitas bulan pertama “Serba” dan bulan kedua “Serba” = $400/600 = 0.667$

Probabilitas bulan pertama “Serba” dan bulan kedua “Ada” = $200/600 = 0.333$

Probabilitas bulan pertama “Ada” dan bulan kedua “Serba” = $250/400 = 0.625$

Probabilitas bulan pertama “Ada” dan bulan kedua “Serba” = $150/400 = 0.375$.

Matriks transisi berdasarkan data tersebut adalah

$$\begin{array}{cc} & \begin{array}{cc} \text{Serba} & \text{Ada} \end{array} \\ \begin{array}{c} \text{Serba} \\ \text{Ada} \end{array} & \begin{bmatrix} 0.667 & 0.333 \\ 0.625 & 0.375 \end{bmatrix} \end{array}$$

Probabilitas untuk toko “Serba” dan “Ada” pada bulan ketiga apabila pada bulan pertama keluarga tersebut memilih untuk berbelanja di toko “Serba” adalah :

$$K_{t(j)} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.667 & 0.333 \\ 0.625 & 0.375 \end{bmatrix} = \begin{bmatrix} 0.667 & 0.333 \end{bmatrix}$$

Probabilitas untuk toko “Serba” dan “Ada” pada bulan ketiga apabila pada bulan pertama keluarga tersebut memilih untuk berbelanja di toko “Ada” adalah :

$$K_{t(j)} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0.667 & 0.333 \\ 0.625 & 0.375 \end{bmatrix} = \begin{bmatrix} 0.627 & 0.375 \end{bmatrix}$$

2.8 Monte Carlo

Monte Carlo merupakan sebuah cara untuk membangkitkan *sample* secara acak pada sebuah permasalahan yang memiliki probabilitas acak. Metode *Monte Carlo* dipopulerkan oleh Stanislaw Marcin Ulam, Enrico Fermi, John von Neumann dan Nicholas Metropolis. Metode ini lalu diberi nama *Monte Carlo* oleh Stanislaw Marcin Ulam atas saran dari Nicholas Metropolis untuk menghormati pamannya yang merupakan seorang penjudi[?]. Nama *Monte Carlo* merupakan nama sebuah kota yang terkenal dengan kasinonya di negara Monako.

Langkah-langkah yang umum dilakukan adalah[?] :

1. Tentukan fungsi distribusi yang akan disimulasikan
2. Tentukan interval angka acak dari setiap variable
3. Pembangkitan angka acak
4. Simulasikan percobaan sebanyak n kali

2.9 Markov Chain Monte Carlo

Markov Chain Monte Carlo merupakan suatu algoritma untuk melakukan prediksi data dari suatu distribusi dengan membangun rantai markov pada suatu distribusi tertentu yang stationer. Pembangkitan data prediksi dilakukan secara acak, namun ada mekanisme penerimaan atau penolakan angka tersebut. Berdasarkan Teorema Bayes, *Posterior* adalah hasil kali antara *Prior* dengan *Likelihood*. *Posterior* adalah probabilitas hipotesa yang ingin kita cari. *Prior* adalah fungsi yang memberikan data sebelum dibuktikan atau probabilitas yang kita yakin akan terjadi sebelum dibuktikan dengan data yang telah terjadi. Distribusi *Prior* bisa saja memiliki

lebih dari satu parameter. *Likelihood* adalah fungsi dari bukti yang ada atau kejadian yang telah terjadi.

Secara umum ada dua cara yang biasa digunakan yaitu *The Metropolis-Hastings Algorithm* dan *Gibbs Sampling*

2.9.1 The Metropolis-Hastings Algorithm

Algoritma *Metropolis-Hastings* menggunakan mekanisme penerimaan dan penolakan (*accept-reject*) untuk membangkitkan barisan data prediksi dari suatu distribusi[7], yang sulit untuk dilakukan penarikan sampel. Pertama adalah melakukan pembangkitan sampel-sample prediksi acak pada distribusi posterior yang diinginkan dengan distribusi normal, lalu hitung rasio probabilitas θ_{new} yaitu data hasil prediksi dengan θ_{t-1} seperti pada persamaan 2.7

$$r(\theta_{new}, \theta_{t-1}) = \frac{Posterior(\theta_{new})}{Posterior(\theta_{t-1})} \quad (2.7)$$

Jika rasio > 1 maka nilai θ_{new} diterima dan jika kurang maka θ_{new} ditolak, lalu membuat angka acak baru dari distribusi seragam, apabila rata-ratanya $< r(\theta_{new}, \theta_{t-1})$ maka angka tersebut diterima menjadi θ_{new} . Proses ini berjalan sebanyak N iterasi.

2.10 PyMC3 Python

PyMC3 merupakan pustaka pemrograman probabilistik berlisensi *open source* yang ditulis dengan bahasa *python*[8]. yang interpretatif, interaktif serta *object-oriented programming* yang berfokus pada keterbacaan kode. Saat ini kode *python* dapat dijalankan pada berbagai macam sistem operasi seperti *windows*, *ubuntu*, *debian*, *machintosh* dan beberapa sistem operasi populer lainnya. *Python* memiliki banyak *library* yang membantu kita dalam analisa data seperti *matplotlib*, *pandas*, *statistic*, *math* dan *pymc*. Dengan fitur-fitur untuk pengolahan data ini *python* menjadi bahasa yang populer dan sering digunakan.

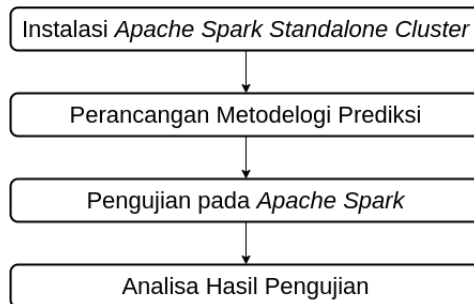
Halaman ini sengaja dikosongkan

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Desain Sistem



Gambar 3.1: Alur kerja penelitian

Penelitian ini bertujuan untuk mengimplementasikan *Apache Spark* untuk memprediksi harga saham menggunakan metode *Markov Chain Monte Carlo* (MCMC). Pembangunan sistem pada penelitian ini berbasis perangkat lunak *Apache Spark* dengan program pengolahan data yang dibuat dengan menggunakan bahasa *python*. Sebelum melakukan analisa mengenai saham, infrastuktur dan sistem dari *Apache Spark* harus dibangun terlebih dahulu. Alur kerja dari penelitian ini ditunjukkan oleh Gambar 3.1.

Sistem yang akan digunakan pada penelitian ini mempunyai lima layer yang ditunjukkan pada Gambar 3.2. Kelima layer tersebut adalah:

1. Layer Infrastruktur

Layer Infrastruktur mencakup dengan virtualisasi, skalabilitas, dan *continuous integration*. Dalam tugas akhir ini, akan dibagun *development environment* di dalam *VirtualBox* yang berisi *virtual machine* yang ditenagai oleh *Apache Spark*. Untuk *scaling*, dapat dibuat *environment* yang mirip pada cloud. Tujuan implementasi *continuous integration* disiapkan untuk mempermudah perpindahan dari *development environment* ke *test* atau *production deployment* jika hasil tugas akhir ini digunakan untuk *production*.

2. Layer *Data Persistence*

Layer ini berfungsi untuk mengatur berbagai penyimpanan sesuai dengan kebutuhan dan bentuk data. Termasuk di dalamnya RDBMS seperti MySQL dan PostgreSQL, penyimpanan data dalam bentuk *key-value* seperti Hadoop dan Redis, basis data kolumner seperti HBase dan Cassandra, dan basis data dokumen seperti MongoDB. Sedangkan format data, misalnya adalah csv, json, atau *plain text*

3. Layer Integrasi

Layer ini fokus kepada akuisisi, transformasi, dan kualitas data dengan menerapkan lima langkah, yaitu:

(a) *Connect*

Menentukan cara menerima data dari berbagai macam sumber, menangani API yang ditentukan oleh sumber data, format input, skema, laju dari pengumpulan data, dan batasan dari penyedia data.

(b) *Correct*

Melakukan transformasi data untuk proses lebih lanjut dan memastikan kualitas dan konsistensi dari data yang diterima tetap terjaga.

(c) *Collect*

Mencari tempat data disimpan dan format data, untuk mempermudah saat langkah *compose* dan *consume*.

(d) *Compose*

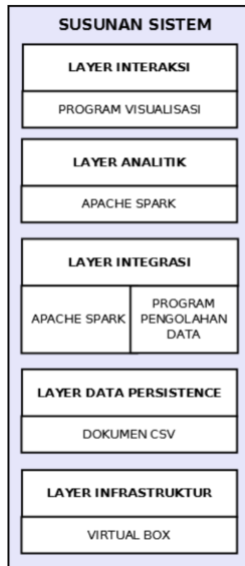
Menentukan cara mengkombinasikan berbagai data set yang telah dikumpulkan dan memperkaya informasi.

- (e) *Consume*
Menentukan penyediaan, pemberian data, dan bagaimana data sampai kepada individu yang tepat di saat yang tepat.
- (f) *Control*
Langkah ini diperlukan karena data terus tumbuh sehingga diperlukan pengaturan pada data.

4. Layer Analitik

Layer ini adalah tempat terjadinya proses pengolahan data menggunakan berbagai macam model, algoritma, dan metode *machine learning* untuk mendapatkan *insight*. Dalam tugas akhir ini, layer analitik ditenagai oleh *Apache Spark*.

5. Layer Interaksi Layer ini melakukan interaksi kepada *end-user* dengan menyediakan *platform* seperti *dashboard* dan visualisasi interaktif.



Gambar 3.2: Rancangan sistem layanan yang digunakan

Kelima layer pada Gambar 3.2 dirancang dengan tujuan untuk memudahkan proses *scaling* dalam mengikuti pertumbuhan dan ragam jenis data, misalnya saat server yang ada sudah tidak mampu mengatasi pengolahan pada data yang ada (kekurangan *resource* memori/cpu) maka dapat ditambahkan server pada layer infrastruktur.

3.2 Data yang Digunakan

Data yang digunakan dalam penelitian ini adalah data historis saham yang didapatkan dari *finance.yahoo.com*. Data yang digunakan termasuk kategori data terstruktur, karena memiliki label dan *field* yang tetap, sehingga dapat dilakukan proses *sort*, *select*, *filter* dan lain sebagainya. Data tersebut didapatkan melalui API Yahoo dengan menggunakan python seperti pada Gambar 3.3.

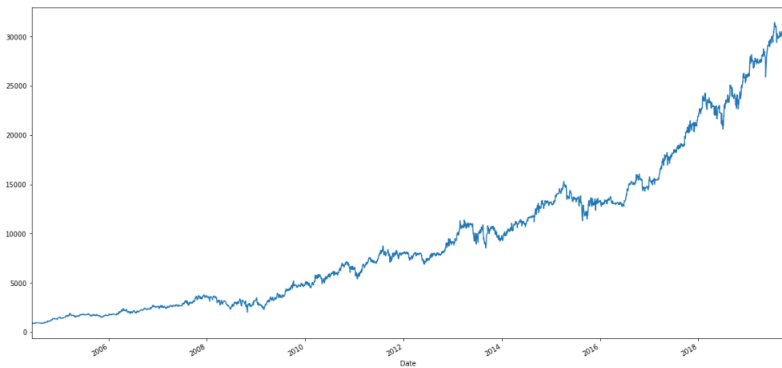
	High	Low	Open	Close	Volume	Adj Close
Date						
2004-06-08	900.0	875.0	875.0	887.5	99830000.0	450.867981
2004-06-09	912.5	875.0	887.5	900.0	58858000.0	457.218384
2004-06-10	900.0	887.5	900.0	900.0	33118000.0	457.218384
2004-06-11	900.0	887.5	887.5	900.0	27166000.0	457.218384
2004-06-14	900.0	875.0	900.0	887.5	31708000.0	450.867981
...
2019-09-25	29675.0	28900.0	29500.0	29675.0	17724000.0	29675.000000
2019-09-26	30300.0	29850.0	30000.0	30300.0	21971800.0	30300.000000
2019-09-27	30375.0	30000.0	30025.0	30350.0	9602000.0	30350.000000
2019-09-30	30350.0	30025.0	30350.0	30350.0	14678200.0	30350.000000
2019-10-01	30325.0	30050.0	30050.0	30200.0	8011400.0	30200.000000

3797 rows × 6 columns

Gambar 3.3: Data saham Bank BCA tahun 2004-2019

Dengan menggunakan API Yahoo, kita juga mengambil data IHSG atau indeks harga saham gabungan pada tiap negara, atau

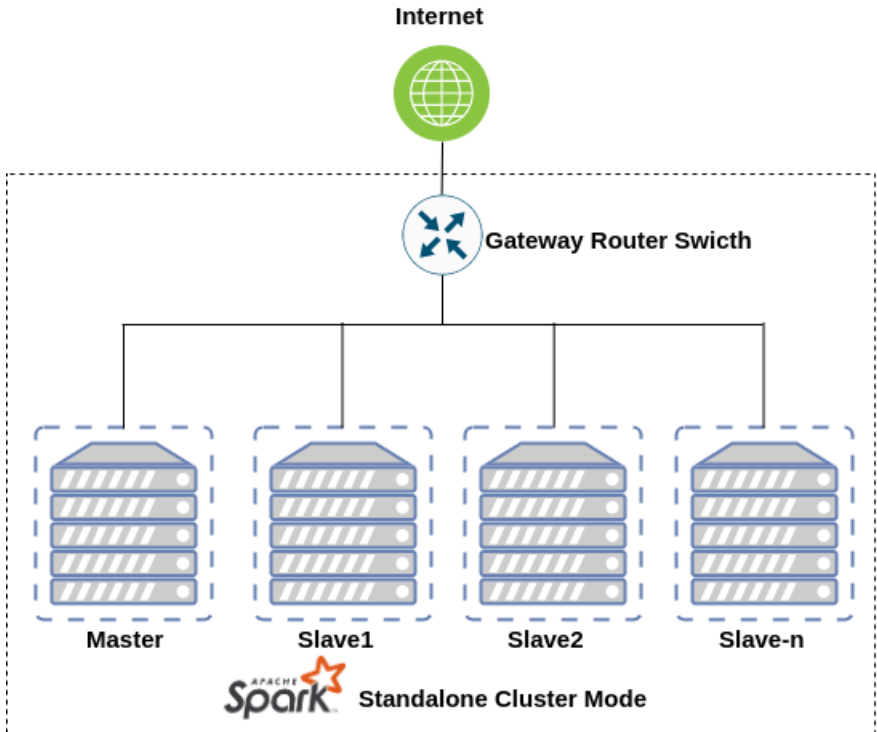
saham-saham perusahaan lain yang terdaftar di pasar sekunder. Data-data yang didapatkan tersebut bersifat *realtime* atau data terbaru pada hari itu. Selain menggunakan API Yahoo data-data tersebut juga bisa didapat menggunakan teknik *crawling* dalam website yang menyediakan data tersebut, namun prosesnya akan lebih sulit. Pada data tersebut kolom yang akan dipakai adalah kolom "Close" lalu sehingga dapat disajikan dalam bentuk grafik seperti pada Gambar 3.4



Gambar 3.4: Pergerakan harga saham

3.3 Pembuatan Apache Spark Multinode Cluster

Sistem dalam penelitian akan di-*deploy* pada komputer server skala lab. Sistem dibuat clustering dengan spark manager sebagai manajemen clusternya, terdiri dari satu *node master* dan enam *node worker/slave* dengan spesifikasi yang berbeda beda. Dalam mode *cluster deployment* pada *Apache Spark*, perlu dilakukan konfigurasi agar *node master* dan *slave* dapat saling berinteraksi. Sebelum melakukan instalasi, pastikan tiap node memiliki sistem operasi yang sama dan *user* yang sama agar lebih mudah saat konfigurasi. Perlu juga dilakukan konfigurasi *passwordless* ssh pada tiap node agar antar node dapat melakukan komunikasi dan eksekusi perintah.



Gambar 3.5: Desain jaringan

3.3.1 Desain Jaringan

Desain jaringan pada penelitian ini menggunakan jaringan privat pada laboratorium Telematika (B201). Alamat IP (*Internet Protocol*) dari *gateway* yang digunakan untuk mengakses jaringan luar adalah 10.122.1.1, berada dalam jaringan 10.122.1.0/24 milik laboratorium Telematika (B201).

Seperti yang terlihat pada Gambar 3.5, sebanyak (n) unit komputer yang terhubung dengan *gateway*. Sistem operasi yang akan digunakan dalam penelitian ini adalah Ubuntu 18.04 (bionic) dengan arsitektur 64-bit pada *master* dan *slaves*. Versi dari Apache Spark yang digunakan adalah versi 2.4.4 yang rilis pada tanggal 30

Tabel 3.1: Alokasi *ip address* tiap *node*

<i>Hostname</i>	<i>IP Address</i>
master	10.122.1.209
slave1	10.122.1.2
slave2	10.122.1.202
slave3	10.122.1.203
slave4	10.122.1.205
slave5	10.122.1.206
slave6	10.122.1.207
slave7	10.122.1.208

Agustus 2019.

Untuk menyiapkan *environment Apache Spark Cluster*, pertama yaitu instalasi sistem operasi yang sama pada setiap node dengan nama *user* yang sama baik node *master* maupun node *slaves*. Pastikan *hostname* dan alokasi *IP address* sesuai seperti Tabel 3.1. Pada penelitian ini saya membuat nama *user* “riset”. Tambahkan juga *hostname* dan *IP address* pada file “/etc/hosts” pada setiap node agar antar node dapat saling mengenali satu sama lain.

Langkah selanjutnya adalah konfigurasi *passwordless ssh* dengan membuat *public key* pada setiap node lalu menyalinnya ke semua node. Hal ini dilakukan agar setiap node dapat berkomunikasi saat program *distributed computing* berjalan. Untuk mengeceknya bisa menggunakan perintah “ssh (hostname)”.

Setelah itu dilanjutkan dengan instalasi program *java* pada setiap node. Pastikan versi programnya sama pada setiap node, karena pada dasarnya *apache spark* dibangun dengan bahasa *java* dan berjalan diatas *java virtual machine*.

Proses selanjutnya adalah instalasi dan konfigurasi *apache spark* pada node master. Untuk program *apache spark* dapat diunduh di halaman resmi *apache spark*. Setelah diunduh lalu ekstrak file ke direktori “/home/riset/spark”. Untuk konfigurasi *apache spark* masuk ke direktori “/home/riset/spark/conf” lalu ubah isi file bernama “spark-env.sh”, lalu tambahkan konfigurasi *IP MASTER HOST* dan direktori *java virtual machine*. Setelah itu ubah isi file “slaves”,

tambahkan *hostname* dari node yang akan dijadikan node *worker/slave*. Setelah semua dilakukan, maka salin direktori “/home/riset/spark” dari node *master* ke semua node *slaves*.

Proses terakhir yaitu tambahkan konfigurasi *SPARK HOME* pada “/home/riset/.bashrc” pada setiap node. Lalu jalankan *Apache Spark Cluster* dengan perintah “start-all.sh” pada node master.

Untuk melakukan pengecekan apakah spark berjalan baik dapat melihat melalui halaman web spark dengan memasukkan IP dan port master pada URL di browser seperti tersaji pada Gambar 3.6 Meskipun memori node slave adalah 1 GB tetapi yang diberikan ke node master kurang dari 1 GB karena dengan asumsi yang lain digunakan untuk sistem. Jika dipaksakan mengalokasikan 2GB untuk diberikan kepada node master, node slave akan error dan harus di-restart


3.4 Perancangan Metodologi Prediksi

Alur metodologi prediksi harga saham adalah seperti pada Gambar 3.7. Yang perlu diperhatikan adalah proses pemilihan data. Sebaiknya data yang digunakan adalah data historis harga saham pada setahun terakhir. mengingat pergerakan saham biasanya mirip dengan kondisi pada tahun sebelumnya.

3.4.1 Akuisisi Data

Pada *API Yahoo Finance*, data disimpan dalam bentuk tabel. Data di perbarui setiap hari kecuali pada hari libur, sehingga perlu dilakukan pengecekan terhadap pembaruan data yang ada di *Yahoo Finance*. Untuk mempermudah Pengecekan dapat dibuat program untuk melakukan pengecekan terhadap data yang ada di *server Yahoo Finance*. Untuk melakukan pengambilan data di *Yahoo Finance* kita harus mengetahui kode perusahaan yang akan kita ambil historis datanya. Misalnya data yang akan diambil adalah data saham PT. Bank Central Asia Tbk, maka kode sahamnya “BBCA.JK”. Kode saham dapat dilihat pada halaman *finance.yahoo.com*. Gambar 3.9 merupakan data saham Bank BCA mulai tanggal 8 Juni 2004 sampai tanggal 31 Desember 2019.

Dari data tersebut lalu dihitung nilai *return* hariannya meng-

 **Spark Master at spark://10.122.1.231:7077**

URL: spark://10.122.1.231:7077

Alive Workers: 3

Cores in use: 3 Total, 0 Used

Memory in use: 3.0 GB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

▼ **Workers (3)**

Worker Id	Address	State	Cores	Memory
worker-20191112172114-10_122.1.232-42559	10.122.1.232:42559	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20191112172114-10_122.1.233-42671	10.122.1.233:42671	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20191112172116-10_122.1.231-38239	10.122.1.231:38239	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

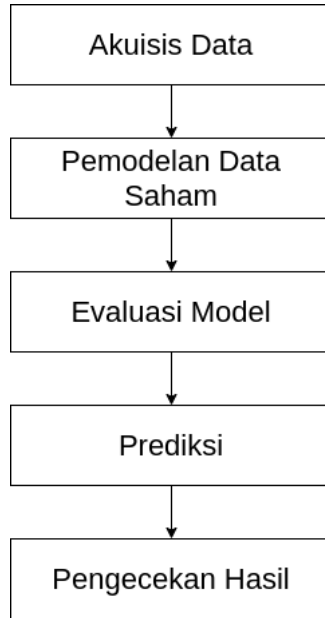
▼ **Running Applications (0)**

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	----------------	------	-------	----------

▼ **Completed Applications (0)**

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	----------------	------	-------	----------

Gambar 3.6: Tampilan *web* untuk memantau *Apache Spark*



Gambar 3.7: Alur implementasi prediksi

gunakan Persamaan 3.3 menjadi seperti pada Gambar 3.10. Data *daily-return* saham inilah data yang akan kita gunakan untuk membuat model prediksi. Setelah data dianalisa, data tersebut kita buat model distribusinya. Cara yang paling mudah adalah dengan melihat grafik histogramnya.

Nilai positif pada kolom *Daily return* data saham pada Gambar 3.10 menunjukkan adanya kenaikan harga saham dibandingkan dengan hari sebelumnya sedangkan nilai negatif menunjukkan adanya penurunan harga saham dibandingkan dengan hari sebelumnya. Dari data didapat nilai rata-rata dari *return* sebesar 0.00093 dengan nilai *return* minimum -0.10622 dan maksimum sebesar 0.12675. Data tersebut dapat divisualisasikan menjadi bentuk grafik pergerakan dari nilai *return* harian dan histogram seperti pada Gambar 3.11 dan Gambar 3.12.

Date	Open	High	Low	Close*	Adj Close**	Volume
Dec 30, 2015	2,077.34	2,077.34	2,061.97	2,063.36	2,063.36	2,367,430,000
Dec 29, 2015	2,060.54	2,081.56	2,060.54	2,078.36	2,078.36	2,542,000,000
Dec 28, 2015	2,057.77	2,057.77	2,044.20	2,056.50	2,056.50	2,492,510,000
Dec 24, 2015	2,063.52	2,067.36	2,058.73	2,060.99	2,060.99	1,411,860,000
Dec 23, 2015	2,042.20	2,064.73	2,042.20	2,064.29	2,064.29	3,484,090,000
Dec 22, 2015	2,023.15	2,042.74	2,020.49	2,038.97	2,038.97	3,520,860,000
Dec 21, 2015	2,010.27	2,022.90	2,005.93	2,021.15	2,021.15	3,760,280,000

Gambar 3.8: Web page of Yahoo Finance

```

Data Saham BBKA.JK Dari 2004-06-08 Sampai 2019-12-31
[*****100%*****] 1 of 1 completed
Date      Open      High      Low      Close     Adj Close  Volume
2004-06-08  875.0    900.0    875.0    887.5     438.253326 99830000
2004-06-09  887.5    912.5    875.0    900.0     444.425873 58858000
2004-06-10  900.0    900.0    887.5    900.0     444.425873 33118000
2004-06-11  887.5    900.0    887.5    900.0     444.425873 27166000
2004-06-14  900.0    900.0    875.0    887.5     438.253326 31708000
...
2019-12-20  33400.0  33700.0  33050.0  33300.0   32734.117188 19890600
2019-12-23  33400.0  33650.0  33250.0  33300.0   32734.117188 9311700
2019-12-26  33300.0  33625.0  33300.0  33400.0   32832.417969 7637700
2019-12-27  33350.0  33500.0  33350.0  33475.0   32906.144531 5989000
2019-12-30  33400.0  33600.0  33350.0  33425.0   32856.992188 12233600
[3867 rows x 6 columns]

```

Gambar 3.9: Pengambilan data saham Bank BCA dari yahoo finance

Setelah dihitung nilai *return*-nya data dibagi menjadi data *training* dan *testing*.

3.4.2 Pemodelan Data Saham

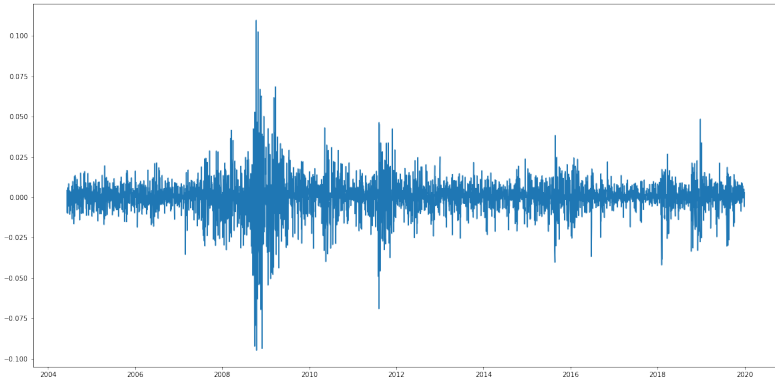
Pada *Apache Spark* data dimodelkan menggunakan *library pyc3* berdasarkan distribusi datanya. Setelah data *daily-return* dianali-

```

Data Saham BBKA.JK Dari 2004-06-08 Sampai 2019-12-31
[*****100%*****] 1 of 1 completed
Date      Open      High      Low      Close      Adj Close      Volume      Daily_returns
2004-06-09  887.5    912.5    875.0    900.0    444.425873    58858000    0.013986
2004-06-10  900.0    900.0    900.0    900.0    444.425873    33118000    0.000000
2004-06-11  887.5    900.0    887.5    900.0    444.425873    27166000    0.000000
2004-06-14  900.0    900.0    875.0    887.5    438.253326    31708000    -0.013986
2004-06-15  875.0    912.5    875.0    912.5    450.598480    59524000    0.027780
...
2019-12-20  33400.0  33700.0  33050.0  33300.0  32734.117188  19890600    0.009050
2019-12-23  33400.0  33650.0  33250.0  33300.0  32734.117188  9311700    0.000000
2019-12-26  33300.0  33625.0  33300.0  33400.0  32832.417969  7637700    0.002999
2019-12-27  33350.0  33500.0  33350.0  33475.0  32906.144531  5989000    0.002243
2019-12-30  33400.0  33600.0  33350.0  33425.0  32856.992188  12233600    -0.001495
[3866 rows x 7 columns]

```

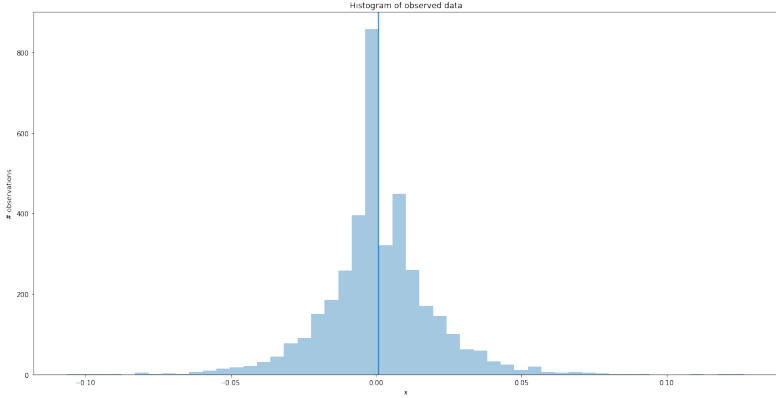
Gambar 3.10: Nilai *daily return* saham Bank BCA



Gambar 3.11: Grafik *daily return* saham Bank BCA

sa dengan melihat histogramnya, data ditentukan model prediksi untun nilai *daily-return*-nya dibuat berdasarkan distribusi datanya. Pada data tersebut ditentukan modelnya menjadi distribusi *Student-T*.

Pada model *likelihood* $P(y|\theta)$ akan dimodelkan sebagai distribusi *StudentT*, dengan model *prior* sebagai parameter [10]. Pada definisi *prior*, *initial value* dapat diubah sesuai dengan derajat kepercayaan pada model data atau tren perusahaan. Pada pembuatan model ini sebenarnya bisa berbeda-beda pada setiap perusahaan, karena setiap perusahaan memiliki strategi yang berbeda-beda yang mempengaruhi naik-turunnya harga saham. Selain itu kondisi



Gambar 3.12: Histogram *daily return* saham Bank BCA

eksternal perusahaan seperti kondisi politik dan krisis juga mempengaruhi. Oleh karena itu sebelum membuat model saham harus dilakukan analisa secara mendalam agar model yang telah dibuat sesuai.

Pada model *prior* $P(\theta)$ data *daily-return* perlu ditentukan juga distribusi setiap parameter yang mempengaruhi model berdasarkan hasil analisa data secara subyektif. Pada distribusi *Student-t* parameter-parameter *probability density function-nya* adalah ν , μ , σ . μ merupakan parameter lokasi data prediksi yang terletak antara -1 sampai 1, σ merupakan parameter volatilitas yang konvergen ke standar deviasi data, ν merupakan parameter normalitas ($\nu > 0$). Dengan kata lain θ merupakan satu atau beberapa parameter yang mempengaruhi distribusi data yang dalam hal ini merupakan data *daily-return* dari saham.

Berdasarkan penelitian yang telah ada sebelumnya, portofolio *return* dari harga saham dimodelkan dengan *geometric Brownian Motion*[9]. Persamaan *geometric Brownian Motion* adalah persamaan diferensial stokastik:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad (3.1)$$

dimana S_t harga saham pada periode t , W_t adalah gerak *brown*

standar, dan μ, σ adalah volatilitas and *drift*. Dengan melakukan pnurunan fungsi persamaannya menjadi:

$$S_t = S_0 \exp((\mu - \sigma/2)t + \sigma W_t) \quad (3.2)$$

Karena pergerakan harga saham mengikuti proses stokastik gerak *brown* maka nilai *return* harga saham harian menjadi:

$$R_t = \ln \left(\frac{S_t}{S_{t-1}} \right) \quad (3.3)$$

dimana R_t adalah nilai *return* harga saham pada waktu t kemudian dihitung rata-ratanya. Berdasarkan metode bayes maka *posterior* \sim *prior* \times *likelihood* maka model *prior*-nya menjadi:

$$\begin{aligned} \log S_t &\sim Normal(\log S_{t-1}, \sigma^2) \\ \sigma &\sim \exp(.5) \\ \mu &\sim Normal(0, 5) \\ v &\sim \exp(.1) \end{aligned} \quad (3.4)$$

```
with pm.Model() as model:
    #prior
    sigma = pm.Exponential('sigma', 1./0.02, testval=.1)
    mu = pm.Normal('mu', 0, sd=5, testval=.1)

    nu = pm.Exponential('nu', 1./10)
    logs = pm.GaussianRandomWalk('logs', tau=sigma**2, shape=n)

    #likelihood
    r = pm.StudentT('r', nu, mu=mu, lam=1/np.exp(-2*logs), observed=returns.values[train])
```

Gambar 3.13: *Data model*

3.4.3 Model Fitting

Data dibagi menjadi data *training* dan data *testing*. Prediksi hanya untuk keesokan hari maka keseluruhan data akan menjadi data *training*. Dilakukan pembangkitan data sampel untuk tiap parameter sebanyak 10000 sampel data dengan panjang rantai sebanyak 500 rantai sesuai panjang data *training*.

Proses pembangkitan sample akan memakan waktu lama karena proses akan berjalan pada prosesor, pada tahap ini fungsi kompu-

```

#Posterior
with model:
  start = pm.find_MAP(vars=[logs], method="L-BFGS-B")

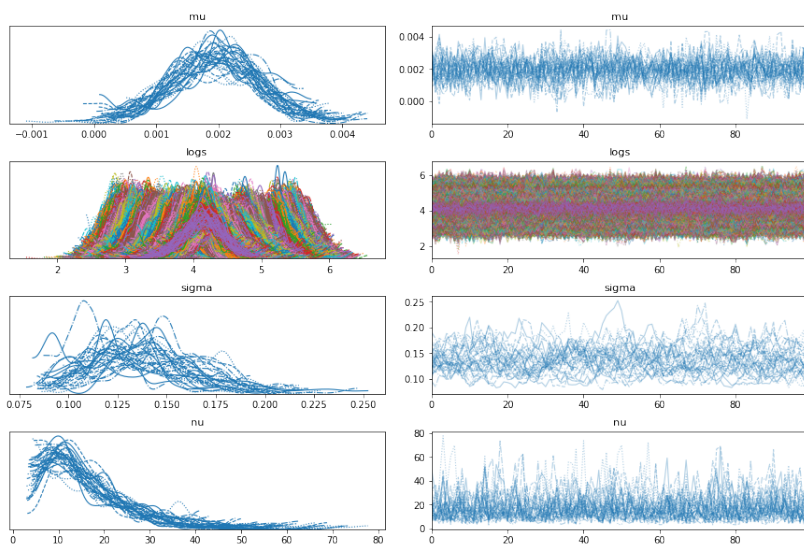
with model:
  step = pm.Metropolis()
  #step = pm.NUTS(vars=[logs, mu, nu, sigma], scaling=start, gamma=.25)
  start2 = pm.sample(5000, step, start=start, progressbar=True, cores=4)[-1]

  # Start next run at the last sampled position.
  step = pm.Metropolis()
  #step = pm.NUTS(vars=[logs, mu, nu, sigma], scaling=start2, gamma=.55)
  trace = pm.sample(10000, step, start=start2, progressbar=True, cores=4)

pm.traceplot(trace)
az.summary(trace)

```

Gambar 3.14: *Data tuning*



Gambar 3.15: *Tracing*

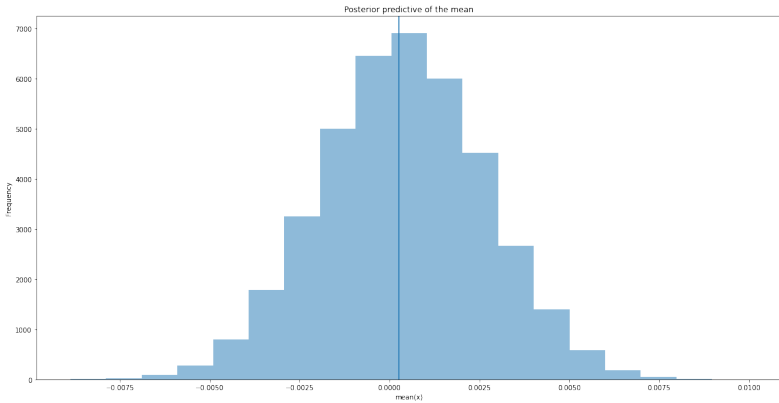
tasi paralel pada *Apache Spark* digunakan. Akan dibandingkan juga waktu pembangkitan sampel menggunakan *single-node* komputer.

	mean	sd	hpd_3%	hpd_97%	mcse_mean	mcse_sd	ess_mean	ess_sd	ess_bulk	ess_tail	r_hat
mu	0.001	0.000	0.000	0.001	0.000	0.000	3617.0	3519.0	3629.0	2718.0	1.01
logs[0]	5.022	0.288	4.446	5.523	0.005	0.003	3794.0	3794.0	3820.0	2712.0	1.01
logs[1]	5.037	0.279	4.543	5.585	0.005	0.003	3510.0	3510.0	3510.0	2880.0	1.00
logs[2]	5.041	0.272	4.518	5.543	0.005	0.003	3641.0	3641.0	3636.0	2589.0	1.01
logs[3]	5.037	0.264	4.559	5.544	0.004	0.003	3546.0	3546.0	3543.0	2753.0	1.01
...
logs[497]	4.499	0.248	4.027	4.950	0.004	0.003	4121.0	4121.0	4141.0	2922.0	1.01
logs[498]	4.511	0.263	4.039	5.019	0.004	0.003	4450.0	4450.0	4446.0	2852.0	1.01
logs[499]	4.522	0.283	3.999	5.066	0.004	0.003	4518.0	4518.0	4536.0	2999.0	1.01
sigma	0.105	0.018	0.072	0.138	0.002	0.002	62.0	62.0	61.0	154.0	1.55
nu	21.686	11.120	5.822	41.471	0.210	0.149	2802.0	2802.0	2737.0	2922.0	1.01

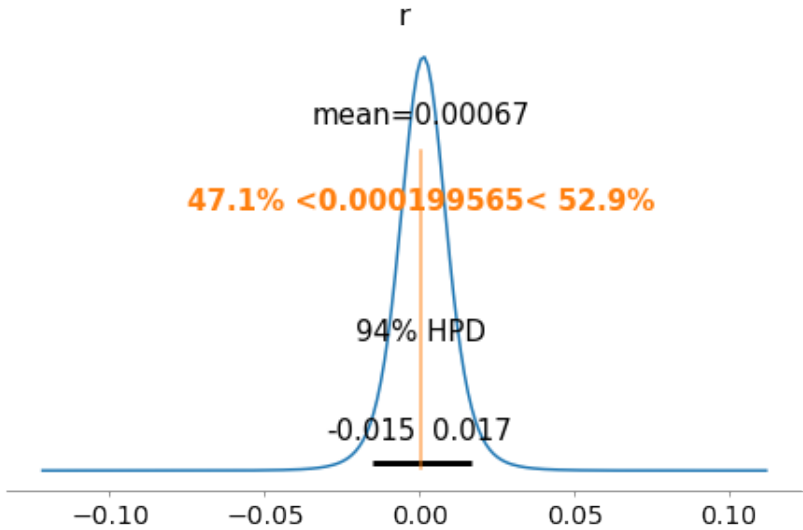
Gambar 3.16: *Tracing Summary*

3.4.4 Posterior Predictive Check

Posterior predictive check atau (PPC) adalah pencarian data dari sampel atau data prediksi yang telah dibangkitkan berdasarkan parameter dari model yang telah dibuat lalu dicek sejauh mana data dari model yang dibuat sesuai dengan data sebenarnya. Pada tahap ini dilakukan penghitungan rasio nilai posterior data prediksi menggunakan algoritma *metropolis-hasting*. Pada tahap ini akan didapatkan analisa hasil dari model prediksi. Untuk mendapatkan



Gambar 3.17: *Posterior predictive check*



Gambar 3.18: *Histogram of posterior predictive check*

hasil prediksi dapat dilakukan dengan rata-rata dari setiap rantai sampel yang dihasilkan, apabila rata-ratanya positif maka harga saham akan naik pada hari berikutnya, apabila rata-ratanya negatif maka harga saham pada hari berikutnya akan turun.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini pengujian dibagi berdasarkan model dan spesifikasi komputer untuk mendapatkan performa waktu komputasi. Pada pengujian model dapat akan didapatkan model prediksi terbaik, serta waktu komputasi pada pembangkitan sampel berdasarkan metode prediksi data. Pada pengujian waktu komputasi dibagi berdasarkan spesifikasi komputer pada *Single Computer*, dan *Apache Spark Multinode Cluster*. Sehingga dengan adanya pengujian tersebut, dapat ditarik beberapa kesimpulan dari pelaksanaan tugas akhir ini.

Adapun perangkat yang digunakan untuk pengujian dari sistem adalah komputer pribadi, komputer Lab B201, dan layanan *Cloud Azure HDInsight Cluster* dengan spesifikasi komputer seperti pada Tabel 4.4, Tabel 4.3 dan Tabel 4.5

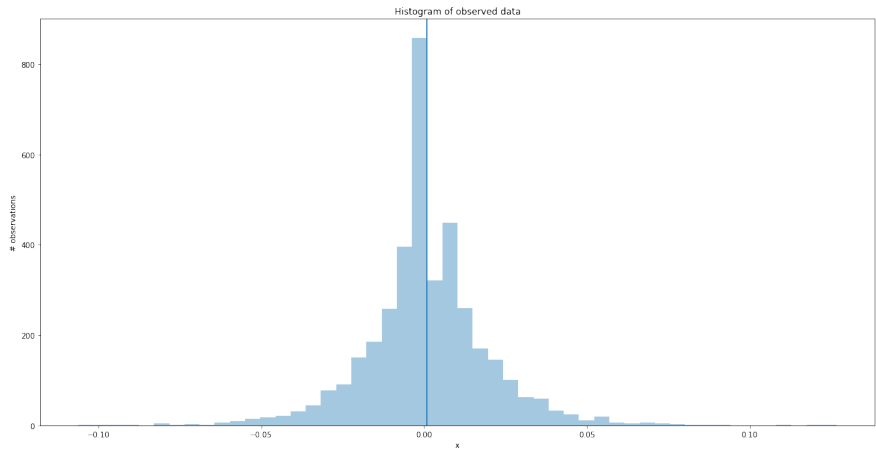
4.1 Evaluasi Model

Pengecekan model dilakukan untuk melihat kesesuaian model kita buat dengan data yang yang diuji. Pengujian ini dilakukan dengan melihat histogram data asli dengan data prediksi lalu dilihat kemiripannya histogramnya. Pada evaluasi model juga dihitung akurasi model. Serta dibandingkan juga waktu komputasi berdasarkan metode prediksi.

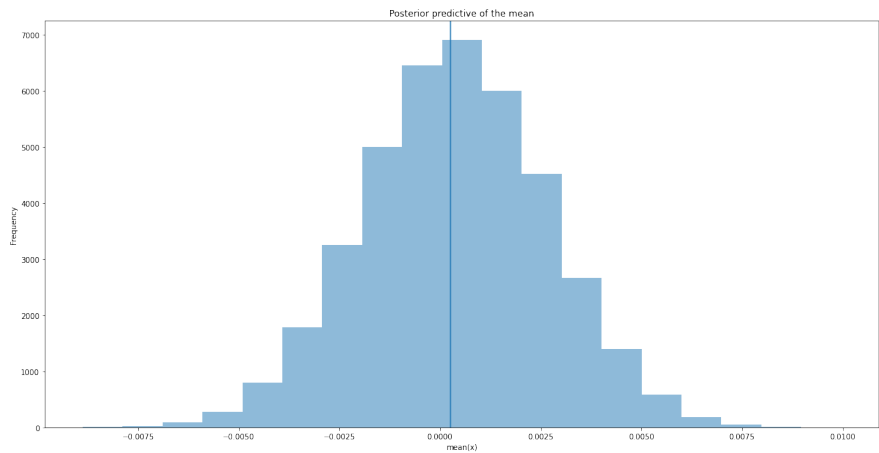
4.1.1 Pengujian Model Student-T distribution

Dengan melihat histogram data asli dengan data prediksi pada Gambar 4.1 dan Gambar 4.2 dapat dilihat bahwa data yang dihasilkan masih kurang mendekati data asli dengan hasil bahwa data yang sesuai hanya sekitar 200 dari 10000 rantai sampel yang dihasilkan untuk setiap parameter. Data yang dihasilkan memiliki rata-rata 0.00067 sedangkan pada data asli rata-ratanya 0.0009. Hasil tersebut dirasa kurang maksimal karena bentuk grafik histogram kurang mendekati model aslinya.

Berdasarkan hasil tersebut dapat dikatakan bahwa data yang cocok hanya sekitar 5%. Dan hasilnya prediksinya adalah P(naik)



Gambar 4.1: Histogram data asli



Gambar 4.2: Histogram data prediksi

sebesar 54.14% dan $P(\text{turun})$ sebesar 45.86%, yang menunjukkan bahwa hasil prediksi untuk bulan januari 2016 adalah harga saham mengalami tren kenaikan. Dari hasil pengujian berdasarkan jumlah data yang digunakan adalah :

Tabel 4.1: Hasil pengujian berdasarkan jumlah data

Jumlah Data	P(naik)	P(turun)
14 Tahun	54.14%	45.86%
2 Tahun	51.62%	48.38%
1 Tahun	50.15%	40.85%
6 Bulan	50.01%	49.99%

Hasil asli pada bulan Januari 2016 adalah sebagai berikut :

4.2 Pengujian Waktu Komputasi pada Single Computer

Waktu komputasi pada model *Student-T* dengan *NUTs sampler* dan *Metropolis sampler* dapat dilihat pada Gambar 4.3 dan Gambar 4.4. Waktu komputasi menggunakan *NUTs sampler* dengan 10000 rantai sampler memakan waktu total 1345 menit atau sekitar 22 jam 30 menit. Sedangkan dengan menggunakan *Metropolis sampler* memakan waktu 18 menit 52 detik. Hasil yang didapatkan baik menggunakan *Metropolis sampler* atau *NUTs sampler* kurang lebih sama yaitu hanya sekitar 5% data yang sesuai.

```

logp = 8,369.9, ||grad|| = 0.061993: 100% | 64/64 [00:00:00:00, 709.78it/s]
Find_MAP_Done
Multiprocess sampling (4 chains in 4 jobs)
CompoundStep
>Metropolis: [logs]
>Metropolis: [nu]
>Metropolis: [mu]
>Metropolis: [sigma]
Sampling 4 chains, 0 divergences: 100% | 42000/42000 [00:54:00:00, 764.44draws/s]
The rhat statistic is larger than 1.4 for some parameters. The sampler did not converge.
The estimated number of effective samples is smaller than 200 for some parameters.
Sampling_Done
Multiprocess sampling (4 chains in 4 jobs)
CompoundStep
>Metropolis: [logs]
>Metropolis: [nu]
>Metropolis: [mu]
>Metropolis: [sigma]
Sampling 4 chains, 0 divergences: 100% | 42000/42000 [01:00:00:00, 693.98draws/s]
The rhat statistic is larger than 1.4 for some parameters. The sampler did not converge.
The estimated number of effective samples is smaller than 200 for some parameters.
Training_Done
100% | 40000/40000 [01:06:00:00, 603.07it/s]
PPC_Done
P(naik): 50.50%
P(turun): 49.50%
P(tetap): 0.0
real 19m41.628s
user 24m20.414s
sys 0m11.015s

```

Gambar 4.3: Waktu komputasi menggunakan *NUTs sampler*

Tabel 4.2: Harga aktual saham GSPC bulan Januari 2016 bersmber dari *finance.yahoo.com*

Tanggal	Harga Aktual
21 Desember 2015	2078,36
22 Desember 2015	2056,50
23 Desember 2015	2060,99
24 Desember 2015	2064,29
28 Desember 2015	2038,97
29 Desember 2015	2021,15
04 Januari 2016	2012,66
05 Januari 2016	2016,71
06 Januari 2016	1990,26
07 Januari 2016	1943,09
08 Januari 2016	1922,03
11 Januari 2016	1923,67
12 Januari 2016	1938,68
13 Januari 2016	1890,28
14 Januari 2016	1921,84
15 Januari 2016	1880,33
18 Januari 2016	1880,33
19 Januari 2016	1881,33
20 Januari 2016	1859,33
21 Januari 2016	1868,99
22 Januari 2016	1906,90
25 Januari 2016	1877,08
26 Januari 2016	1903,63
27 Januari 2016	1882,95
28 Januari 2016	1893,36
29 Januari 2016	1940,24

Tabel 4.3: Spesifikasi perangkat pengujian pada laboratorium Telematika B201 Teknik Komputer ITS

Hardware/Software	Specification
CPU	Core i7-9700F 3.00GHz
CPU Cores	8 Cores
L3 cache	12 MB
Memory	32 GB
Operating System	Ubuntu 18.04

Tabel 4.4: Spesifikasi perangkat pengujian menggunakan laptop pribadi

Hardware/Software	Specification
CPU	Core i5-2450M 2.50GHz
CPU Cores	4 Cores
L3 cache	3 MB
Memory	12 GB
Operating System	Kali Linux 2020b

Tabel 4.5: Spesifikasi perangkat pengujian menggunakan layanan Cloud Service

Hardware/Software	Specification
CPU	Xeon 8272CL 2.60GHz
CPU Cores	16 Cores
L3 cache	36 MB
Memory	128 GB
Operating System	Ubuntu 16.04

Tabel 4.6: Waktu pengujian pada *Apache Spark*

Mode	Jumlah Data	Waktu
Single 2-cores	10000 rantai	138 menit
Single 4-cores	20000 rantai	253 menit
Cluster 16-cores	10000 rantai	57 menit
Cluster 16-cores	20000 rantai	93 menit

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil implementasi dan pengujian model yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut :

1. Kesesuaian model yang dibuat dengan data asli hanya 5% sehingga hasil prediksi selalu berada di angka kurang lebih 50%. Pembuatan model yang digunakan membutuhkan analisa fundamental saham, karena tren dan strategi perusahaan juga dapat mempengaruhi.
2. Pembuatan model dan pengecekan model membutuhkan waktu yang cukup lama, karena perlu dilakukan berulang kali pembuatan dan pengecekan model agar didapatkan model yang sesuai/maksimal. Model saham setiap perusahaan dapat berdeda beda.
3. Jumlah node mempengaruhi waktu pembangkitan rantai sampel. Semakin banyak node yang dipakai maka komputasi akan semakin cepat karena jumlah processor yang dipakai semakin banyak. Kemampuan processor juga berpengaruh karena pembangkitan rantai berada pada node prosesor.
4. Jumlah memori hanya mempengaruhi jumlah maksimum rantai sampel yang dihasilkan. Semakin banyak *resource* memori yang ada jumlah maksimum rantai yang dapat dibangkitkan semakin banyak.
5. *Apache Spark* memiliki keterbatasan *library* untuk *machine learning* sehingga untuk dibutuhkan modifikasi pada program yang dibuat agar dapat berjalan di *Apache Spark*.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut :

1. Sebelum membuat model saham harus diperhatikan analisa pada kondisi eksternal perusahaan sebagai pembanding pada

hasil prediksi.

2. Pembuatan model harus diperhatikan agar hasil prediksi tepat, perlu dilakukan pengecekan model berulang kali agar data yang dihasilkan sesuai atau mendekati data asli.
3. Diperlukan penambahan memori dan node agar dapat komputasi dapat lebih cepat, sehingga waktu yang dibutuhkan untuk membuat dan mengecek model tidak cukup lama.
4. Diperlukan prosesor dengan kemampuan tinggi atau yang memiliki jumlah *core* banyak serta memori *register cache* pada prosesor yang besar agar komputasi lebih cepat.

DAFTAR PUSTAKA

- [1] I. Gibranata, The Detective of Stock. Surabaya, Jawa Timur: YLT Publishing, 2011. (Dikutip pada halaman 1, 10).
- [2] F. Nelli, Python Data Analytics. Berkeley, CA New York: Apress, Distributed to the Book trade worldwide by Springer, 2015. (Dikutip pada halaman 5).
- [3] A. Garg, “Apache Spark Architecture,” 2019. (Dikutip pada halaman 8).
- [4] “Bayes’ theorem.” https://en.wikipedia.org/wiki/Bayes27_theorem. Terakhir diakses pada tanggal 22 November 2019. (Dikutip pada halaman 14).
- [5] “Geometric brownian motion.” https://en.wikipedia.org/wiki/Geometric_Brownian_motion. Terakhir diakses pada tanggal 22 November 2019. (Dikutip pada halaman 15).
- [6] “Student’s t-distribution.” https://en.wikipedia.org/wiki/Student27s_t-distribution. Terakhir diakses pada tanggal 22 November 2019. (Dikutip pada halaman 15).
- [7] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Handbook of Markov Chain Monte Carlo. CRC press, 2011. (Dikutip pada halaman 19).
- [8] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, “Probabilistic programming in Python using PyMC3,” PeerJ Computer Science, vol. 2016, no. 4, 2016. (Dikutip pada halaman 19).
- [9] D. Brigo, A. Dalessandro, M. Neugebauer, and F. Triki, “A Stochastic Processes Toolkit for Risk Management,” SSRN Electronic Journal, pp. 1–43, 2011. (Dikutip pada halaman 33).
- [10] M. D. Hoffman and A. Gelman, “The No-U-Turn Sampler : Adaptively Setting Path Lengths in Hamiltonian Monte Carlo,” vol. 15, pp. 1351–1381, 2014. (Dikutip pada halaman 32).

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Alfin Alim Muhammad lahir pada 22 November 1995 di Surabaya, Jawa Timur. Penulis lulus dari SMP Negeri 1 Surabaya pada tahun 2011 kemudian melanjutkan pendidikan di SMA Negeri 5 Surabaya hingga lulus pada tahun 2014. Penulis kemudian melanjutkan pendidikan sarjana ke Departemen Teknik Komputer ITS Surabaya pada bidang studi Telematika. Saat di kuliah penulis aktif dalam berbagai organisasi baik di dalam maupun di luar kampus termasuk Himpunan Mahasiswa Teknik Elektro ITS dan Ikatan

Alumni SMA Negeri 5 Surabaya. Saat di kuliah penulis aktif menjadi Wakil Ketua Tim Kaderisasi HIMATEKTRO ITS 2016/2017 dan Wakil Ketua Tim Event HIMATEKTRO ITS 2016/2017. Penulis juga aktif menjadi Asisten laboratorium B201 (Telematika) hingga saat ini. Selama masa kuliah penulis aktif dalam, aktif dalam *development group networking* dan juga sebagai *administrator* jaringan Lab B201. Penulis sangat tertarik dengan segala hal yang berhubungan dengan komputer, dan berencana mendalami cabang ilmu komputer lain selain jaringan komputer.

Halaman ini sengaja dikosongkan