

SKRIPSI

STUDI DAN IMPLEMENTASI SPARK STREAMING UNTUK
MENGUMPULKAN *BIG DATA STREAM*



Muhammad Ravi

NPM: 2016730041

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2020

UNDERGRADUATE THESIS

**STUDY AND IMPLEMENTATION OF SPARK STREAMING
TO COLLECT BIG DATA STREAM**



Muhammad Ravi

NPM: 2016730041

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2020**

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

STUDI DAN IMPLEMENTASI SPARK STREAMING UNTUK MENGUMPULKAN *BIG DATA STREAM*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 9 Juni 2020



Muhammad Ravi
NPM: 2016730041

LEMBAR PENGESAHAN

**STUDI DAN IMPLEMENTASI SPARK STREAMING UNTUK
MENGUMPULKAN *BIG DATA STREAM***

Muhammad Ravi

NPM: 2016730041

Bandung, 9 Juni 2020

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Pascal Alfadian, Nugroho, M.Comp.

Natalia, M.Si.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

STUDI DAN IMPLEMENTASI SPARK STREAMING UNTUK MENGUMPULKAN *BIG DATA STREAM*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 9 Juni 2020

Muhammad Ravi
NPM: 2016730041

ABSTRAK

Big Data bukan hanya tentang data dengan ukuran yang besar. Tetapi, data juga dihasilkan dengan sangat cepat. Seiring perkembangan teknologi, internet, dan pengguna internet, data datang dari berbagai macam sumber yang berbeda-beda dengan format yang berbeda-beda pula. Selain itu data harus bisa ditransformasi dengan cepat untuk meningkatkan analisis dan keputusan bisnis. Sehingga data tidak bisa diolah secara konvensional lagi; menyimpan seluruh data dengan format yang berbeda pada *data lake*.

Dengan menggunakan konsep-konsep pengolahan data stream, proses ekstraksi data dari sumber, transformasi format data, dan penyimpanan data bisa dilakukan secara *real-time*. Konsep-konsep data stream akan diimplementasikan dengan *Apache Spark*, sebuah *framework* untuk mengolah data secara terdistribusi. Untuk mengolah data stream, *Apache Spark* akan menggunakan API *spark streaming* yang mengumpulkan data dengan potongan-potongan kecil (*windows*) dan dengan *Spark SQL* yang menambahkan aliran data sebagai baris baru pada tabel, *Structured Streaming*. Selain itu *Apache Spark* akan diintegrasikan dengan teknologi lain, Kafka, sistem yang digunakan untuk menghubungkan sistem-sistem lain dari berbagai sumber data dan meningkatkan performa *streaming*.

Penelitian ini dilakukan untuk mengumpulkan data stream dan membandingkan performa *Spark Streaming* ketika mengumpulkan data stream secara langsung dengan *Structured Streaming* yang terintegrasi dengan Kafka. Eksperimen dilakukan untuk melihat performa *Spark Streaming* dan *Structured Streaming* dalam mengumpulkan data stream.

Langkah-langkah penelitian untuk *Spark Streaming* adalah dengan mempelajari konsep data stream, mempelajari teknologi *Apache Spark* dan *Spark Streaming*, menginstal *Apache Spark* pada kluster dan mengeksplorasi dengan menggunakan API *Spark Streaming*; membuat *Dstream*; dan menghitung kata-kata yang masuk pada sistem; mengumpulkan data *twitter* secara real-time dengan *library* yang disediakan oleh *Spark*.

Langkah-Langkah penelitian untuk *Structured Streaming* hampir sama dengan *Spark Streaming*. Tetapi, ada langkah-langkah tambahan yaitu mempelajari teknologi *Apache Kafka* beserta konsepnya, menginstal Kafka dan Zookeeper pada kluster, mengeksplorasi dengan membuat topik, menentukan partisi, dan membuat konektor, dan mengintegrasikan Kafka dengan *Structured Streaming*. Adapun, data yang dikumpulkan oleh Kafka ada dua yaitu data dari *twitter* dan data cuaca yang keduanya dikumpulkan oleh Kafka Connect sebelum ditulis ke topik dan diambil oleh *Structured Streaming*.

Bentuk dari perangkat lunak kedua eksperimen ini akan berupa jar-jar yang akan dieksekusi di kluster dan performanya bisa dilihat di antar muka web dan hasil eksekusi akan disimpan di HDFS

Kesimpulan pertama yang didapat dari penelitian ini adalah *Structured Streaming* jauh lebih baik dalam mengumpulkan data stream dibandingkan *Spark Streaming* karena *Structured Streaming processing time* maka pengumpulan jadi lebih cepat. Kesimpulan kedua yang didapat adalah, *Kafka* jauh lebih baik untuk mengumpulkan data stream karena bisa mengumpulkan data stream dari berbagai sumber dengan cepat dan *fault tolerant*. Tetapi, *Structured Streaming* baik dalam mentransformasi data tidak terstruktur menjadi terstruktur dan ekstraksi fitur.

Kata-kata kunci: *Big Data, Data Stream, Apache Spark, Spark Streaming, Structured Streaming, Kafka*

ABSTRACT

Big Data is not only about size. But also, data is coming really fast. As the internet and its user gradually grow, data comes in different shapes and sources. Data must be transformed quickly, so we can get insights and make business decision faster. Thus, data cannot be computed in a conventional way by dumping all of the data in data lake.

By using Streaming concepts, extracting data from sources, transforming data format, and loading the data can be done in real-time manner. Streaming Concepts will be implemented using Apache spark, cluster-computing framework that compute data in a distributed way. Apache Spark will use API, spark streaming that collect data by making small batches called window and Spark SQL treating each data stream as a row that continuously appended into larger table. Structured Streaming will also be integrated to another system, kafka, that connecting many source and target systems and handling data transfer between system.

This research is done by collecting Data stream and then comparing the performance of spark streaming and structured streaming that integrated with kafka. The procedures taken to do this research are studying streaming concept such as processing time;event time;and windowing, studying cluster-computing framework such as *Apache Spark* and *Apache Kafka*, installing *Apache Spark* on cluster environment and exploring Spark Streaming by creating Dstream and counting words, installing Apache Kafka and exploring it by creating topics, set a partition, and creating connectors. The result of this research are executable jars that run on clusters. The output, performance graph and result, of this jars can be seen on a web interface.

In conclusion, structured streaming is far more efficient than spark streaming. Due to its lack of processing time. Thus, it compute data faster and more in real-time manner. Moreover, structured streaming is well integrated with another system and library such as MLlib and Kafka. When it comes to collecting data Kafka is better than spark or structured streaming because kafka is written and save in binary, it can collecting data from many system at once and different format. But, when it comes to transforming data and feature extraction structured streaming is better because it can deconstruct complex data format.

Keywords: Big Data, Data Stream, Apache Spark, Spark Streaming, Structured Streaming, Kafka

Untuk Ibu dan Ayah. . .

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Kuasa karena telah mengatur seluruh kejadian, kesempatan, dan hal-hal lain yang di luar kendali penulis berada pada pihak penulis. Sehingga, skripsi ini dapat selesai dan kerja keras yang selama ini penulis lakukan membuahkan hasil. Pada Kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua yang telah membiayai kuliah selama ini
2. Ibu Veronica Sri Moertini selaku dosen pembimbing yang telah banyak membantu, mengarahkan, dan memberi masukan selama proses pembuatan skripsi ini
3. Pak Kristopher David Harjono yang selalu bersedia menjawab pertanyaan jika ada kesulitan dan pelajaran *Data Mining*-nya yang sangat membantu.
4. Teman-teman seperjuangan kuliah dan skripsi
5. Orang-orang yang saya temui di internet yang membuat video dan artikel untuk menerangkan konsep-konsep
6. pihak-pihak lain yang terlalu banyak untuk disebutkan satu per satu yang telah membantu skripsi ini

Akhir kata, penulis menyadari bahwa skripsi ini tidak lepas dari kekurangan dan penulis berharap skripsi ini bisa membantu pembelajaran dan penelitian selanjutnya

Bandung, Juni 2020

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE PROGRAM	xxviii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Big Data	5
2.2 Big Data Stream	7
2.2.1 Pengertian Stream Processing	7
2.2.2 Pemodelan Stream Processing	8
2.2.3 Pola Bounded Processing	8
2.2.4 Pola Unbounded Processing	11
2.2.5 Arsitektur Stream Processing	13
2.3 Sistem Terdistribusi Hadoop	15
2.3.1 Arsitektur Hadoop	16
2.3.2 Hadoop Distributed File System(HDFS)	16
2.3.3 MapReduce	17
2.4 Scala	19
2.4.1 variabel	20
2.4.2 Fungsi	20
2.4.3 Kelas	21
2.4.4 Kelas Option	22
2.4.5 Trait	23
2.4.6 Tuple	23
2.4.7 Koleksi	23
2.4.8 Percabangan	25
2.4.9 Pengulangan	25
2.4.10 Operasi Baca Tulis File	26
2.5 Sistem Terdistribusi Spark	27
2.5.1 Susunan Spark	28

2.5.2	Application Programming Interface (API) Spark	29
2.5.3	Arsitektur Apache Spark	30
2.6	Spark Streaming	31
2.6.1	Arsitektur Spark Streaming	33
2.6.2	Transformasi Spark Streaming	34
2.6.3	Output Operations	35
2.6.4	Checkpointing	35
2.7	Structured Streaming	36
2.7.1	Cara Kerja Structured Streaming	36
2.7.2	Event Time Windowing	38
2.7.3	Mengatasi Data Terlambat	38
2.8	Kafka	41
2.8.1	Messages and Batches	42
2.8.2	Schemas	42
2.8.3	Topics	42
2.8.4	Brokers	43
2.8.5	Producers	44
2.8.6	Consumers	44
2.8.7	Zookeeper	45
3	STUDI EKSPLORASI	47
3.1	Konfigurasi Klaster	47
3.1.1	Konfigurasi Hadoop	47
3.1.2	Konfigurasi Spark	48
3.2	Konfigurasi API dan Data Collector	49
3.2.1	Konfigurasi TCP Socket	49
3.2.2	Konfigurasi Twitter API	49
3.2.3	Konfigurasi Kafka Standalone	50
3.2.4	Konfigurasi Kafka Cluster	52
3.3	Studi Eksplorasi	53
3.3.1	Eksplorasi Spark Streaming dengan TCP Socket	53
3.3.2	Eksplorasi dengan Twitter API	55
3.3.3	Eksplorasi Kafka Cluster	56
4	ANALISIS DAN PERANCANGAN	63
4.1	Analisis Data Stream Twitter dengan Spark Streaming	63
4.1.1	Pengumpulan Data Stream	63
4.1.2	Analisis Data Stream	63
4.1.3	Analisis Masukan dan Keluaran	64
4.1.4	Rancangan Arsitektur	65
4.2	Analisis Data Stream Twitter dengan Structured Streaming	70
4.2.1	Pengumpulan Data Stream dengan Kafka	71
4.2.2	Analisis Data Stream	74
4.2.3	Analisis Masukan dan Keluaran	76
4.2.4	Rancangan Arsitektur	76
4.3	Analisis Data Stream Cuaca dengan Structured Streaming	76
4.3.1	Pengumpulan Data Stream	77
4.3.2	Analisis Data	80
4.3.3	Rancangan Arsitektur	82
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	83
5.1	Implementasi Perangkat Lunak	83

5.1.1	Lingkungan Perangkat Keras	83
5.1.2	Lingkungan Perangkat Lunak	84
5.2	Pengujian Spark Streaming Twitter	84
5.2.1	Eksperimen 1	85
5.2.2	Eksperimen 2	87
5.2.3	Perbandingan Hasil Eksperimen	95
5.3	Pengujian Structured Streaming Twitter	96
5.3.1	Performa Structured Streaming	96
5.3.2	Keluaran Eksekusi	98
5.4	Pengujian Pengumpulan Data Cuaca	98
5.4.1	Pengujian Kafka Connect Source	98
5.4.2	Pengujian Structured Streaming	101
5.4.3	Pengujian Kafka Connect Sink	102
6	KESIMPULAN DAN SARAN	105
6.1	Kesimpulan	105
6.2	Saran	106
	DAFTAR REFERENSI	107
	A KODE PROGRAM EKSPLORASI	109
	B KODE PROGRAM KAFKA CONNECT	113
	C KODE PROGRAM SPARK STREAMING TWITTER	123
	D KODE PROGRAM STRUCTURED STREAMING TWITTER	127
	E KODE PROGRAM STRUCTURED STREAMING CUACA	133

DAFTAR GAMBAR

2.1	Gambar Pemetaan <i>Time-Domain</i>	8
2.2	Gambar <i>fixed-window</i>	9
2.3	Gambar <i>Session-batch</i>	9
2.4	Gambar <i>Filtering</i>	10
2.5	Gambar <i>Inner-join</i>	10
2.6	Gambar <i>approximation-algorithm</i>	10
2.7	Gambar <i>Windowing</i>	11
2.8	Gambar <i>Watermarks</i>	13
2.9	Gambar Arsitektur <i>Lambda</i>	14
2.10	Gambar Arsitektur <i>Hadoop</i>	16
2.11	Gambar Arsitektur HDFS	17
2.12	Gambar Arsitektur MapReduce	18
2.13	Gambar Proses MapReduce	19
2.14	Gambar <i>Spark unified stack</i>	28
2.15	Gambar Arsitektur <i>Spark</i>	30
2.16	Gambar Arsitektur <i>Spark Streaming</i>	32
2.17	Gambar Alur <i>Spark Streaming</i>	32
2.18	Gambar Alur <i>Dstream</i>	33
2.19	Gambar mengubah <i>Data Stream</i> dari lines ke words	33
2.20	Gambar eksekusi <i>Spark Streaming</i> pada komponen <i>Spark</i>	34
2.21	Gambar cara kerja <i>Windowed Transformation</i>	34
2.22	Gambar tabel Structured Streaming	36
2.23	Gambar proses Structured Streaming	37
2.24	Gambar Event-time windowing	38
2.25	Gambar data terlambat structured-streaming	39
2.26	Gambar watermark structured-streaming update-mode	40
2.27	Gambar watermark structured-streaming append-mode	41
2.28	Gambar Publisher/Subscriber	42
2.29	Gambar Topic pada Kafka	43
2.30	Gambar stream topic	43
2.31	Gambar Kafka Broker	44
2.32	Gambar Zookeeper	45
2.33	Gambar arsitektur	46
2.34	Gambar quorum	46
3.1	Gambar Instalasi Java	48
3.2	Gambar HADOOPHOME	48
3.3	Gambar Spark	49
3.4	Gambar Twitter Tokens	50
3.5	Gambar Instalasi Java	51
3.6	Gambar ZOOKEEPER_HOME	51
3.7	Gambar menjalankan server	51

3.8	Gambar menjalankan topic	52
3.9	Gambar <i>consumer-producer</i>	52
3.10	Gambar File input	53
3.11	Gambar pengaturan spark streaming	54
3.12	Gambar perhitungan url	54
3.13	Gambar File input	54
3.14	Gambar Output Web Log	55
3.15	Gambar Setup Twitter	55
3.16	Gambar Setup Spark Streaming	55
3.17	Gambar transformasi twitter	56
3.18	Gambar folder output	56
3.19	Gambar file output	56
3.20	Gambar file output	56
3.21	Gambar Halaman Zookeeper CLI	57
3.22	Gambar Halaman utama Zoonavigator	57
3.23	Gambar Pengaturan Zoonavigator	58
3.24	Gambar Kafka CLI	58
3.25	Gambar Kafka CLI list	59
3.26	Gambar Kafka CLI delete topic	59
3.27	Gambar Kafka CLI List setelah dihapus	59
3.28	Gambar Kafka Manager List Topic	60
3.29	Gambar Kafka Manager Create Topic	60
3.30	Gambar Kafka CLI membuat producer	61
3.31	Gambar Kafka CLI membuat consumer	62
3.32	Gambar Kafka CLI membuat onsumer-from-beginning	62
4.1	Gambar Flowchart Trending Topics	64
4.2	Gambar Arsitektur Streaming	65
4.3	Gambar alur streaming context	65
4.4	Gambar Transformasi Dstream	66
4.5	Gambar RDD Partition	66
4.6	Gambar RDD Batch Size	66
4.7	Gambar RDD chain transformation	67
4.8	Gambar RDD Windowing	67
4.9	Gambar Key exchange	68
4.10	Gambar sorting RDD	68
4.11	Gambar RDD Save	69
4.12	Gambar RDD waiting	69
4.13	Gambar RDD waiting	70
4.14	Gambar Twitter Request	71
4.15	Gambar Twitter Object	72
4.16	Gambar diagram kelas Kafka	73
4.17	Gambar Arsitektur Structured Streaming	76
4.18	Gambar General Architecture Kafka	77
4.19	Gambar Class Digaram Kafka	77
4.20	Gambar Kafka Connect Architecture	78
4.21	Gambar Kafka Cluster	79
4.22	Gambar Data Kafka	79
4.23	Gambar Simulasi Kafka	80
4.24	Gambar Arsitektur Pengumpulan Data	82
5.1	Gambar Eksekutor Spark Streaming	85

5.2	Gambar Statistkik Spark Streaming	86
5.3	Gambar Statistik Spark Streaming	86
5.4	Gambar Hasil Spark Streaming	87
5.5	Gambar Eksekutor Spark Streaming 2	88
5.6	Gambar Statistik Streaming 5 jam	89
5.7	Gambar Delay Streaming 5 jam	89
5.8	Gambar Statistik Streaming selama 10 jam	90
5.9	Gambar Delay Spark Streaming 10 jam	90
5.10	Gambar Hasil Spark Streaming	91
5.11	Gambar Trending Topic Setiap 30 menit	92
5.12	Spark Streaming	93
5.13	Spark Streaming	94
5.14	Gambar Perkembangan Mention	95
5.15	Gambar Performa Structured Streaming 1	97
5.16	Gambar microbatch structured streaming 1	97
5.17	Gambar DarkSkySource Connector	99
5.18	Gambar Konfigurasi Kafka Connect	100
5.19	Gambar Kumpulan Connector untuk setiap Kota	100
5.20	Gambar Topologi kumpulan connector	101
5.21	Gambar Kafka Connect Sink	102
5.22	Gambar Sink Connector Elasticsearch	103
5.23	Gambar Elasticsearch topografi	103
5.24	Gambar Antarmuka Elasticsearch	104
5.25	Gambar Penyajian data pada Elasticsearch	104

DAFTAR TABEL

4.1	DataFrame Hasil Transformasi	74
4.2	DataFrame Hasil Agregasi Windowing	75
4.3	DataFrame Hasil Agregasi Windowing	75
5.1	Hashtag yang sering digunakan selama 10 jam	92
5.2	Hashtag yang sering digunakan selama 10 jam	94
5.3	Perbandingan Hasil Eksperimen	95
5.4	Perbandingan Hasil Eksperimen Cuaca	101

DAFTAR KODE PROGRAM

2.1	Deklarasi Variable	20
2.2	Deklarasi Fungsi	20
2.3	Deklarasi Fungsi Ringkas	20
2.4	Contoh fungsi <i>high-order</i>	20
2.5	Contoh fungsi anonim	21
2.6	Penulisan ringkas fungsi anonim	21
2.7	Penggunaan fungsi anonim pada fungsi <i>high-order</i>	21
2.8	Contoh definisi kelas	21
2.9	Pembuatan instansi kelas	21
2.10	Contoh definisi kelas singleton	22
2.11	Definisi <i>case class</i>	22
2.12	Pembuatan instansi <i>case class</i>	22
2.13	Fungsi pengubah String menjadi Int	22
2.14	Contoh pemanfaatan kelas Some dan None	23
2.15	Contoh penggunaan trait	23
2.16	Deklarasi dan penggunaan tuple	23
2.17	Contoh penggunaan array	24
2.18	Macam-macam cara pembuatan list	24
2.19	Contoh penggunaan Vector	24
2.20	Contoh deklarasi set	24
2.21	Contoh deklarasi dan penggunaan map	24
2.22	Contoh dasar percabangan	25
2.23	Penggunaan else if dalam percabangan	25
2.24	Contoh dasar penggunaan for	25
2.25	Iterasi pada array	25
2.26	Pengulangan angka secara terurut menurun	25
2.27	Pengulangan bertingkat	26
2.28	Pengulangan menggunakan while	26
2.29	Penggunaan do-while	26
2.30	Membaca file teks	26
2.31	Menulis file teks	26
2.32	contoh checkpointing	35
3.1	zookeeper.properties	53
3.2	zookeeper	53
4.1	Input Twitter	64
4.2	Output Twitter	65
A.1	URLCounter.scala	109
A.2	SaveTweets.scala	109
A.3	HashtagsCounter.scala	110

A.4	ErrorCounter.scala	110
A.5	Utilities.scala	111
A.6	zookeeper.properties	112
B.1	DarkSkySourceTask.java	113
B.2	DarkSkyHttpClient.java	114
B.3	DarkSkySourceConnector.java	116
B.4	DarkSkySourceConnectorConfig.java	116
B.5	DarkSkySchema.java	117
B.6	LatitudeValidator.java	119
B.7	LongitudeValidator.java	119
B.8	RequestValidator.java	119
B.9	TimestampValidator.java	119
B.10	BatchSizeValidator.java	120
B.11	Weather.java	120
C.1	Main.scala	123
C.2	TwitterTask.scala	123
C.3	/Utilities.scala	125
D.1	Main.scala	127
D.2	TwitterTask.scala	127
D.3	TwitterSchema.scala	129
D.4	Utilities.scala	131
E.1	Main.scala	133
E.2	DarkSkySchema.scala	133
E.3	DarkSkyTask.scala	134
E.4	Utilities.scala	135

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Data Berkembang sangat cepat dalam beberapa tahun terakhir. Semakin banyak orang yang terhubung ke perangkat internet, semakin banyak juga akses terhadap website dan media sosial. Selain itu, sensor-sensor perangkat elektronik yang saling terhubung ke internet semua meninggalkan jejak digital berupa data. Data yang terakumulasi ini berukuran besar dengan format yang bervariasi dan aliran data yang datang sangat cepat. Big data akan lebih mudah untuk dianalisis ketika seluruh datanya telah terkumpul dan direduksi. Reduksi data adalah pengecilan ukuran data dengan mengambil rangkuman dari sekelompok data. Tetapi, masalah muncul ketika aliran data datang secara cepat, terus menerus, dan data tersebut harus diolah secara *real-time* untuk mendapatkan informasi. Masalah-masalah yang muncul tersebut antara lain; bagaimana data tersebut bisa ditangkap, dianalisis, dan disimpan.

Teknik yang digunakan untuk menangkap dan mengumpulkan aliran big data berbeda dari teknik mengumpulkan big data biasa karena pola aliran data berbeda, aliran data berupa potongan-potongan data yang datang secara terus-menerus tidak seperti big data biasa yang berupa file-file ukuran besar sekaligus.

Aliran data tersebut bisa langsung diolah dan dianalisis, banyak informasi-informasi bermanfaat yang bisa didapat. Contohnya, data bisa menjadi bahan pertimbangan untuk pengambilan keputusan bisnis. Tetapi, tidak semua data memiliki nilai dan sifat yang sama. Ada data yang memiliki nilai lebih ketika bisa langsung dianalisis ketika didapatkan. Kebutuhan untuk langsung mendapatkan dan menganalisis data secara *real-time* menjadi sangat penting. Selain itu, teknik pengumpulan data yang digunakan untuk pola data yang datang secara terus menerus berbeda dengan teknik yang digunakan untuk mengumpulkan dan mengolah data biasa. *Big Data* yang perlu diakses secara *real-time* adalah page views pada sebuah website, sensor pada IoT (*Internet of Things*).

Selain itu kebutuhan untuk mengolah data dengan cepat semakin penting karena nilai suatu data cenderung menurun seiring bertambahnya waktu. Banyak Perusahaan dan Organisasi yang membutuhkan data untuk diolah secara cepat. Semakin cepat data bisa diambil, dianalisis, dimanipulasi, dan semakin banyak throughput yang bisa dihasilkan maka sebuah organisasi akan lebih *agile* dan responsif. Semakin sedikit waktu yang digunakan untuk ETL (*Extract, Load, Transform*) pekerjaan akan semakin fokus untuk melakukan analisis bisnis.

Untuk menjawab masalah di atas, *Spark Streaming* merupakan teknologi yang menjadi salah satu solusi terhadap adanya kebutuhan untuk menganalisis *big data* secara *real time*. Data hasil streaming kemudian dapat dianalisis dengan teknik-teknik komputasi sederhana dan divisualisasikan agar lebih mudah dimengerti.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijabarkan, berikut adalah rumusan masalah untuk skripsi ini.

1. Bagaimana Karakteristik *data stream* dan contoh-contoh analisisnya?

2. Bagaimana cara kerja *Spark Streaming*?
3. Bagaimana cara mengintegrasikan *Spark Streaming* untuk mengumpulkan data?

1.3 Tujuan

Berikut ini adalah tujuan yang ingin dicapai pada skripsi ini.

1. Melakukan studi tentang definisi, teknik pengumpulan, arsitektur, dan manfaat analisis dari data stream
2. Mempelajari konsep, arsitektur, cara kerja *Spark Streaming* dan integrasinya dengan teknologi-teknologi lain
3. Mengimplementasikan *Spark Streaming* pada sebuah sistem untuk mengumpulkan data stream dengan kasus-kasus tertentu.

1.4 Batasan Masalah

Batasan masalah untuk skripsi ini adalah sebagai berikut.

1. Data yang diolah bisa berubah dan memiliki batasan akses sesuai penyedia data tersebut
2. Data akan diolah secara terdistribusi pada 10 komputer saja.
3. Pengembangan hanya fokus di ekstraksi data stream dan analisis data stream agregasi saja tidak sampai pembuatan model *machine learning*

1.5 Metodologi

Metodologi yang digunakan dalam pembuatan skripsi ini adalah sebagai berikut.

1. Mempelajari pola, arsitektur, dan sumber dari *Big Data Stream*.
2. Mempelajari arsitektur, cara kerja, dan komponen-komponen *Spark*.
3. Mempelajari Distribusi data pada *Hadoop distributed file System*.
4. Mempelajari arsitektur dan cara kerja *Spark Streaming* pada *Spark*.
5. Mempelajari Bahasa pemrograman *Scala*.
6. Mempelajari *Kafka* dan *Structured Streaming*.
7. Mengimplementasikan *Spark Streaming* pada kluster.
8. Mengintegrasikan *Structured Streaming* dengan *Kafka*.
9. Melakukan eksperimen dan pengujian *Spark Streaming* dan *Structured Streaming* pada kluster

1.6 Sistematika Pembahasan

Sistematika penulisan skripsi ini adalah sebagai berikut.

1. Bab Pendahuluan
Bab 1 membahas tentang latar belakang, rumusan masalah, tujuan, Batasan masalah, metodologi penelitian, dan sistematika pembahasan.
2. Bab Landasan Teori
Bab 2 membahas tentang teori-teori mengenai *Big Data*, *Big Data Stream*, Sistem terdistribusi *Spark*, *Spark Streaming*, dan *Kafka*.
3. Bab Studi Eksplorasi
Bab 3 membahas tentang langkah-langkah untuk melakukan konfigurasi kluster pada *hadoop*, konfigurasi kluster untuk *Spark* dan *kafka*, hasil studi eksplorasi *Spark Streaming*.
4. Bab Analisis dan Perancangan
Bab 4 membahas tentang analisis perangkat lunak *Spark*, analisis data uji, analisis masukan dan keluaran, rancangan Arsitektur, dan diagram kelas.
5. Bab Implementasi dan Eksperimen
Bab 5 membahas tentang implementasi perangkat lunak, eksperimen performansi dan analisis hasil eksperimen.
6. Bab kesimpulan dan Saran
Bab 6 membahas tentang kesimpulan yang disampaikan penulis setelah melakukan penelitian ini dan saran-saran untuk pengembangan lanjut.