ANALISA *BIG DATA* PADA *CLUSTER* KOMPUTER MENGGUNAKAN KOMPUTASI TERDISTRIBUSI

OLEH:

ZAINUDIN

09011181924004

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2023

ANALISA *BIG DATA* PADA *CLUSTER* KOMPUTER MENGGUNAKAN KOMPUTASI TERDISTRIBUSI

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

OLEH:

ZAINUDIN
09011181924004

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2023

LEMBAR PENGESAHAN

ANALISA *BIG DATA* PADA *CLUSTER* KOMPUTER MENGGUNAKAN KOMPUTASI TERDISTRIBUSI

TUGAS AKHIR

Program Studi Sistem Komputer
Jenjang S1

OLEH:

ZAINUDIN
09011181924004

Indralaya, 7 November 2023

Mengetahui,

Ketua Jurusan Sistem Komputer	Pembimbing Tugas Akhir
Dr. Ir. Sukemi, M.T.	**Ahmad Heryanto, S.Kom, M.T.**
NIP. 196612032006041001	**NIP. 198701222015041002**

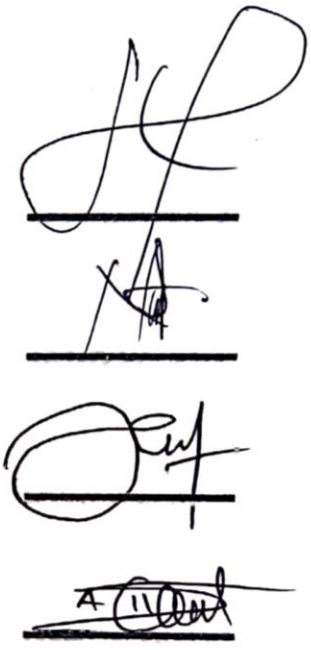
HALAMAN PERSETUJUAN

Telah diuji dan lulus pada:

Hari : Senin
Tanggal : 2 Oktober 2023

Tim Penguji :

1. **Ketua Sidang : Huda Ubaya, S.T., M.T.**
2. **Sekretaris Sidang : Nurul Afifah, S.Kom., M.Kom.**
3. **Penguji Sidang : Ahmad Fali Oklilas, S.T., M.T.**
4. **Pembimbing : Ahmad Heryanto, S.Kom., M.T.**



Mengetahui,

Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Zainudin

NIM : 09011181924004

Judul : ANALISA *BIG DATA* PADA *CLUSTER* KOMPUTER MENGGUNAKAN KOMPUTASI TERDISTRIBUSI

Hasil pengecekan *software Ithenticate/Turnitin* : 2%

Menyatakan bahwa Laporan Skripsi saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam Laporan Skripsi ini, maka sata bersedia menerima sanksi akademik dari Universitas Sriwijaya. Demikian, pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan

Indralaya, 15 November 2023

Zainudin

NIM. 09011181924004

HALAMAN PERSEMBAHAN

Saya persembahkan tugas akhir ini kepada orang yang peduli akan pendidikan akademis dan moral yaitu

Ibu dan Bapak Tercinta

Nabi SAW dalam sebuah riwayat dari Ibnu Umar Radiyallahu `Anhu (RA): Aku datang menemui Nabi Muhammad SAW pernah bersama 10 orang. Kemudian salah seorang anshar bertanya, **siapakah orang yang paling cerdas dan mulia, wahai rasulullah?**

Mendengar pertanyaan tersebut, Rasulullah pun menjawab, "**orang yang paling cerdas ialah dia yang paling banyak mengingat kematian serta paling siap menghadapinya**. Mereka itulah orang-orang cerdas, sebab mereka akan pergi membawa kemuliaan dunia dan kehormatan" (HR. Ibnu Majah).

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "**Analisa *Big Data* pada *Cluster* Komputer menggunakan Komputasi Terdistribusi**". Tugas Akhir ini disusun sebagai syarat untuk melengkapi salah satu syarat memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Sriwijaya.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada beberapa pihak atas ide dan saran serta bantuannya dalam menyelesaikan penulisan Tugas Akhir ini. Oleh karena itu, penulis ingin mengucapkan rasa syukur kepada Allah SWT dan terima kasih kepada yang terhormat :

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan penulisan Tugas Akhir ini dengan baik dan lancar.
2. Ibu saya Hermina, S.Ag. dan Ayah saya Lukman Hakim yang selalu senantiasa mendoakan dan mendukung saya dalam menyelesaikan Tugas Akhir ini.
3. Bapak Prof. Dr. Erwin, S.Si., M.Si. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. H. Sukemi, M.T. selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Bapak Ahmad Heryanto, S.Kom., M.T. selaku Dosen Pembimbing Tugas Akhir yang telah berkenan meluangkan waktunya guna membimbing, memberikan saran dan motivasi serta bimbingan terbaik untuk penulis dalam menyelesaikan Tugas Akhir ini.
6. Bapak Prof. Deris Stiawan, M.T., Ph.D., IPU., ASEAN ENG., CPENT. selaku dosen Pembimbing Akademik.
7. Mbak Renny selaku admin Jurusan Sistem Komputer yang telah membantu mengurus seluruh berkas.
8. Teman-teman COMNETS yang telah membersamai dan membantu

dalam pengerjaan Tugas Akhir ini.

9. Teman-teman CALMPONK ada bayu (borju), hendi (kepin), rahman (mamang), ocim, habib (bedul), angga (alek), agung (mamang kenten), nauvan dimas, dan vano yang telah menemani selama masa-masa kuliah dan saling meluangkan waktu untuk berkeluh kesah dimasa penulisan Tugas Akhir.
10. Teman Kosan ada imam, raja, dan ijal yang telah menyemangati serta memotivasi saya dikala dalam mengerjakan Tugas Akhir ini.
11. Teman-teman Komala Residence yang telah menghibur dan melakukan banyak aktivitas dalam masa-masa penulisan Tugas Akhir ini.
12. Teman-teman KANTIN BUNDA ANJAS yang membantu menghibur dan menemani dikala bosan dan jenuh dalam pengerjaan Tugas Akhir ini.
13. Dan semua pihak yang telah membantu.

Penulis mengharapkan dan membuka diri untuk segala kritik dan saran yang membangun dari semua pihak sebagai acuan untuk penulisan Tugas Akhir yang lebih baik lagi. Akhir kata kami ucapkan banyak terima kasih kepada semua pihak yang telah membantu. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca sekalian.

Indralaya, 15 November 2023

Penulis,

Zainudin

NIM. 09011181924004

BIG DATA ANALYSIS ON COMPUTER CLUSTERS USING DISTRIBUTED COMPUTING

Zainudin (09011181924004)
Department of Computer Systems, Faculty of Computer Science, Sriwijaya University
Email: zainudin2001@gmail.com

ABSTRACT

Along with the development of the era of globalization, the use of technology has been very widespread in various industrial sectors, so data accumulates in a very fast time to grow into large-scale data called big data. The emergence of big data makes the formulation of optimization problems more complicated, because of the large volume and complexity of the data, therefore it is necessary to implement a parallel and distributed computer cluster architecture. There are several methods that support parallelization and computing systems to perform data processing such as MPI (Message Processing Interface), OpenMP (Open Multi Processing), Hadoop, Spark, and others. In the context of big data, many data structures in big data become more complex, high dimensions, and large sizes. This study utilizes the parallelization system of the Apache Spark framework system which is used as a medium to conduct distributed computer clusters to carry out big data processing. The results of this study showed that the distributed cluster system on spark effectively read big data, in the wordcount experiment on 31,788,324 rows of data, spark was faster with a time difference of 84.6 seconds. The performance produced in the spark library, MLlib, to conduct machine learning classification experiments and recommendation system to carry out advanced big data processing, the performance produced in the classification model gets the best value with an accuracy of 94.95%, F1-score 95%, recall 95.18%, and precision 94.77% of the 6 models used, while for the recommendation system with Algorithm ALS (Alternating Least Squares) got an RMSE score of 0.46 from 5 experiments with different tune parameters.

Keywords: *Distributed Computing, Big Data, Computer Clusters, Apache Spark*

ANALISA *BIG DATA* PADA *CLUSTER* KOMPUTER MENGGUNAKAN KOMPUTASI TERDISTRIBUSI

Zainudin (09011181924004)
Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya
Email: zainudin2001@gmail.com

ABSTRAK

Seiring berkembangnya era globalisasi, pemanfaatan teknologi sudah sangat luas diberbagai sektor industri, sehingga data terakumulasi dalam waktu yang sangat cepat pertumbuhannya menjadi data yang berskala besar yang disebut dengan *big data.* Munculnya *big data* ini membuat perumusan masalah optimasi menjadi lebih rumit, karena volume yang besar dan kompleksitas pada data tersebut, maka dari itu perlu menerapkan arsitektur *cluster* komputer paralel dan terdistribusi. Terdapat beberapa metode yang mendukung sistem paralelisasi dan komputasi untuk melakukan pemrosesan data seperti MPI *(Message Processing Interface), OpenMP (Open Multi Processing),* Hadoop, Spark, dan lainnya. Dalam konteks *big data*, banyak struktur data pada *big data* menjadi lebih kompleks, dimensi yang tinggi, dan ukuran yang besar. Pada penelitian ini memanfaatkan sistem paralelisasi dari sistem *framework* apache spark yang digunakan sebagai media untuk melakukan *cluster* komputer yang terdistribusi untuk melakukan pemrosesan *big data*. Hasil penelitian ini menunjukan sistem *cluster* terdistribusi pada spark efektif membaca *big data*, pada percobaan *wordcount* terhadap 31.788.324 baris data, spark lebih cepat dengan selisih waktu 84.6 detik. Performa yang dihasilkan pada *library* spark yaitu MLlib, untuk melakukan percobaan *machine learning classification* dan r*ecommendation system* untuk melakukan pengolahan *big data* lanjutan, performa yang dihasilkan pada model klasifikasi mendapat nilai terbaik dengan akurasi 94.95%, F1-score 95%, recall 95.18%, dan presisi 94.77% dari 6 model yang digunakan, sedangkan untuk *recommendations system* dengan Algoritma *ALS (Alternating Least Squares) mendapat* nilai RMSE 0.46 dari 5 percobaan dengan tune parameter yang berbeda.

Kata Kunci: Komputasi Terdistribusi, *Big Data*, *Cluster* Komputer, Apache Spark

DAFTAR ISI

LEMBAR PENGESAHAN iii

HALAMAN PERSETUJUAN iv

HALAMAN PERNYATAAN..... v

HALAMAN PERSEMBAHAN vi

KATA PENGANTAR..... vii

ABSTRACT..... ix

ABSTRAK x

DAFTAR ISI..... xi

DAFTAR GAMBAR..... xv

DAFTAR TABEL xviii

BAB I PENDAHULUAN..... 1

1.1 Latar Belakang.....1

1.2 Perumusan Masalah.....3

1.3 Batasan Masalah.....3

1.4 Tujuan.....3

1.5 Manfaat.....4

1.6 Metodologi Penelitian 4

1.7 Sistematika Penulisan.....5

BAB II TINJAUAN PUSTAKA..... 6

2.1 Penelitian Terkait.....6

2.2 Big Data 11

2.2.1 Tipe *Big Data*.....12

2.2.2 Karakteristik *Big Data* 12

2.3 Cluster Computing 14
2.4 Komputasi Terdistribusi *(Distributed Computing)* 14
2.5 RDD *(Resilient Distributed Datasets)* 15
2.6 Apache Spark 16
2.6.1 Spark Streaming 17
2.6.2 Spark MLlib 17
2.6.3 Spark GraphX 19
2.6.4 Spark SQL 20
2.7 Machine Learning Classification 21
2.7.2 Decision Tree 21
2.7.3 Naïve Bayes 21
2.7.4 Logistic Regression 22
2.7.5 Support Vector Machine 22
2.7.6 Gradient Boosted Tree (GBT) 23
2.8 Alternating Least Squares (ALS) 24
2.9 RMSE (Root Mean Squared Error) 24
2.10 Confusion Matrix 24
2.10.1 Akurasi 25
2.10.2 Presisi 26
2.10.3 Recall 26
2.10.4 F1-score 27
BAB III METODOLOGI PENELITIAN 28
3.1 Kerangka Kerja 28
3.1.1 Studi literatur 30
3.1.2 Kebutuhan Perangkat Keras 30
3.1.3 Kebutuhan Perangkat Lunak 30

3.1.4 Melakukan Instalasi dan Pengaturan Apache Spark 31
3.1.5 Pengumpulan Datasets 31
3.1.6 Melakukan Konfigurasi *Cluster Manager* 32
3.1.6.1 Membuat *Master Machine* 33
3.1.6.2 Menjalankan *Worker Machine* 33
3.1.6.3 Menjalankan PySpark pada Notebook 33
3.1.7 Membuat Session pada Spark 34
3.1.8 Percobaan *WordCount* dan *Load* Dataset 34
3.1.9 Pra Pengolahan 34
3.1.10 Membuat Data Training dan Testing 35
3.1.11 Perancangan Model dan Menyesuaikan Data 35
3.1.12 Pengujian Model Spark MLlib 35
3.2 Sistem Komputasi Terdistribusi pada Apache Spark 36
3.3 Skenario Pengujian dan Percobaan 38
3.3.1 Percobaan Penggunaan Sample Data 38
3.3.2 Percobaan *Wordcount* 39
3.3.3 Percobaan Penerapan *Machine Learning Classification* 41
3.3.3.1 Percobaan Algoritma Naïve Bayes 42
3.3.3.2 Percobaan Algoritma Logistic Regression 42
3.3.3.3 Percobaan Algoritma Decision Tree 43
3.3.3.4 Percobaan Algoritma Random Forest 43
3.3.3.5 Percobaan Algoritma Support Vector Machine (SVM) 43
3.3.3.6 Percobaan Algoritma Gradient Boosted Trees (GBT) 44
3.3.4 Percobaan Penerapan *Recommendation Engine* 44
3.3.5 Pengujian Spark yang Terdistribusi 45
3.4 Analisis 47

3.5 Kesimpulan.....47

BAB IV HASIL DAN PEMBAHASAN 48

4.1 Pendahuluan 48

4.2 Pengaturan Apache Spark pada Sistem Operasi.....48

4.3 Konfigurasi *Cluster* Komputer yang Terdistribusi.....50

4.4 Wordcount pada Data.....54

4.5 Pra Pengolahan Data.....57

4.6 Evaluasi Hasil Percobaan 69

4.7 Analisa Hasil Secara Keseluruhan.....80

BAB V KESIMPULAN DAN SARAN 82

5.1 Kesimpulan.....82

5.2 Saran.....83

DAFTAR PUSTAKA 84

DAFTAR GAMBAR

Gambar 2. 1 Karakteristik Big data [22] 13

Gambar 2. 2 Sebuah Sistem Komputasi Terdistribusi [25].....15

Gambar 2. 3 Spark Structured Streaming Sncremental Execution [27].....17

Gambar 2. 4 Fitur Utama MLlib.....18

Gambar 2. 5 Logistic Regression Apache Spark dan Hadoop [27].....19

Gambar 2. 6 Performa End-to-end PageRank (20 iterations, 3.7B edges) [27] ...20

Gambar 2. 7 Confusion Matrix.....25

Gambar 3. 1 Kerangka Kerja Penelitian.....29

Gambar 3. 2 Diagram Blok Percobaan Penelitian.....29

Gambar 3. 3 Komponen Cluster Spark [7].....32

Gambar 3. 4 Tahapan Melakukan Konfigurasi Cluster Komputer.....33

Gambar 3. 5 Struktur Spark Session.....34

Gambar 3. 6 Skenario Komputasi Terdistribusi pada Apache Spark.....36

Gambar 3. 7 Struktur RDD.....37

Gambar 3. 8 Flowchat Diagram Wordcount 39

Gambar 3. 9 Tahapan Percobaan Wordcount pada Pyspark 41

Gambar 3. 10 Struktur Persiapan Data.....45

Gambar 3. 11 Struktur Uji Coba Spark 46

Gambar 4. 1 File Biner Hadoop.....49

Gambar 4. 2 Tampilan Membangun Master Machine Pada Spark.....52

Gambar 4. 3 Menghubungkan Worker Machine pada Spark 52

Gambar 4. 4 Tampilan WebUI pada apache spark.....53

Gambar 4. 5 Tampilan untuk Mengintegrasikan Spark pada Notebook 54

Gambar 4. 6 Running Program dipyspark.....54

Gambar 4. 7 Deskripsi Tahapan Job Worker Machine 55

Gambar 4. 8 Ringkasan Matriks Job Selesai.....55

Gambar 4. 9 Jumlah Kolom dan Baris 57

Gambar 4. 10 Tampilan Data Articles.....57

Gambar 4. 11 Grafik Penjualan Articles 58

Gambar 4. 12 Tampilan Data Customers 58

Gambar 4. 13 Data Pembelian Customers.....59

Gambar 4. 14 Tampilan Data Transactions.....59

Gambar 4. 15 Grafik Jumlah Transactions.....60

Gambar 4. 16 Mark Artikel Terbaru.....60

Gambar 4. 17 Urutan Tertinggi Data Customers.....61

Gambar 4. 18 Urutan Tertinggi Jumlah Transaksi pada Artikel 62

Gambar 4. 19 Tampilan setelah Proses String Indexer 63

Gambar 4. 20 Jumlah Kolom dan Baris 63

Gambar 4. 21 Tampilan Data test_data 64

Gambar 4. 22 Tampilan Data train_data 64

Gambar 4. 23 Jumlah Data train dan test.....65

Gambar 4. 24 Penggabungan Data 65

Gambar 4. 25 Memerika nilai pada kolom.....65

Gambar 4. 26 Pengecekan Data Class.....65

Gambar 4. 27 Rebalance Data Class 66

Gambar 4. 28 Grafik Korelasi Dataset 66

Gambar 4. 29 Drop Kolom 67

Gambar 4. 30 Tampilan Data setelah Proses Encoding.....67

Gambar 4. 31 Tampilan setelah Normalisasi Data 68

Gambar 4. 32 Confusion Matrix pada Naïve Bayes 70

Gambar 4. 33 Classification Report Naïve Bayes 70

Gambar 4. 34 Confusion Matrix pada Logistic Regression 71

Gambar 4. 35 Classification Report Logistic Regression.....72

Gambar 4. 36 Confusion Matrix pada Decision Tree.....73

Gambar 4. 37 Classification Report Decision Tree 73

Gambar 4. 38 Confusion Matrix pada Ramdom Forest.....74

Gambar 4. 39 Classification Report Random Forest 75

Gambar 4. 40 Confusion Matrix pada Support Vector Machine.....76

Gambar 4. 41 Classification Report Support Vector Machine 76

Gambar 4. 42 Confusion Matrix pada Gradient-Boosted Trees (GBT) 77

Gambar 4. 43 Classification Report Gradient-Boosted Trees 78

Gambar 4. 44 Grafik Nilai RMSE 79

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait 6

Tabel 3. 1 Spesifikasi Perangkat Keras 30

Tabel 3. 2 Spesifikasi Perangkat Lunak 31

Tabel 3. 3 Data Pengujian Wordcount 40

Tabel 4. 1 Path untuk Environment Variabels.....50

Tabel 4. 2 Command untuk Membangun Cluster Spark 51

Tabel 4. 3 Hasil Percobaan Wordcount.....56

Tabel 4. 4 Hasil dari Model Naïve Bayes.....69

Tabel 4. 6 Hasil dari Model Logistic Regression.....71

Tabel 4. 8 Hasil dari Model Decision Tree 72

Tabel 4. 10 Hasil dari Model Random Forest 74

Tabel 4. 12 Hasil dari Model Support Vector Machine 75

Tabel 4. 14 Hasil dari Model Gradient-Boosted Trees (GBT).....77

Tabel 4. 16 Tabel Rincian Hasil ALS 79

Tabel 4. 17 Hasil Percobaan Spark MLlib Classification 80

Tabel 4. 18 Hasil Percobaan Spark MLlib Recommendation Enginee.....81

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya era globalisasi, pemanfaatan teknologi informasi sudah sangat luas di berbagai industri baik itu di Indonesia ataupun dunia. Seperti yang dijelaskan pada penelitian (Hamidreza Kadkhodaei et al., 2021) bahwa saat ini laju pertumbuhan data telah berkembang sangat pesat hampir setiap dua tahunnya, volume data di dunia di berbagai industri menjadi dua kali lipat, sehingga data terakumulasi dalam waktu yang sangat cepat pertumbuhannya menjadi data yang berskala besar yang disebut *big data*. Karena volume yang sangat besar dan kompleksitas pada data tersebut, metode pemrosesan data yang biasa, tidak cocok untuk menanganinya, sehingga tidak memungkinkan untuk menyimpan dan memproses data dengan jumlah yang banyak pada satu komputer saja [1].

Munculnya data berskala besar ini membuat perumusan masalah optimasi menjadi lebih kompleks. Untuk mengatasi masalah tersebut, perlu menerapkan prosedur iteratif dalam lingkungan komputasi terdistribusi [2]. Pertumbuhan ukuran data dan keberagaman struktur data ini tidak dapat dengan mudah untuk dianalisis dan membutuhkan lebih banyak waktu dan sumber data yang kompleks untuk di proses [3]. Maka dari itu perlu menerapkan arsitektur *cluster* komputer paralel dan terdistribusi.

Berdasarkan penelitian (J. Rayes-Ortiz et al) terdapat beberapa metode yang mendukung sistem paralelasi dan komputasi untuk melakukan pemrosesan data, seperti MPI *(Message Processing Interface),* OpenMP *(Open Multi Processing),* Hadoop, Spark dan lainnya. Performa yang hasilkan pada komputasi paralel ini sangat bergantung pada jenis masalah dan jenis arsitektur sistem yang digunakan. Dalam konteks *big data,* Algoritma optimasi yang biasa digunakan mungkin tidak lagi efektif, karena saat ini banyak struktur data pada *big data* menjadi lebih kompleks, dimensi yang tinggi, dan ukuran yang besar, data tersebut memiliki karakteristik yang mencakup sejumlah data terstruktur, tidak terstruktur, ataupun

semi terstruktur, sehingga akurasi dan kinerja algoritma pemrosesan data biasa menurun secara signifikan [4]. Hadoop dan Spark dirancang untuk melakukan pengolahan *big data*, namun sebagai pengolahan data yang berskala besar *framework* komputasi pada apache spark lebih cocok untuk menangani data yang besar karena memproses data secara *in-memory* [5]. Apache spark ditulis dalam bahasa pemrograman Scala dan juga menyediakan API. Apache spark mendukung berbagai bahasa pemrograman yaitu Scala, Java, Python, dan R [6], terdapat *library* dalam ekosistem Spark yaitu Spark SQL, Spark MLlib, Spark GraphX, dan Spark Streaming, sehingga memberikan perluang tambahan untuk analisis data yang berskala besar.

Pada penelitian sebelumnya [7], membahas tentang analisis *big data* pada e-governance menggunakan apache spark. Penelitian ini membuktikan bahwa spark dapat membantu dalam menganalisis data yang dikumpulkan oleh pemerintah secara akurat dengan kecepatan tinggi, yang menunjukan waktu eksekusi untuk melakukan analisis menggunakan spark-shell pada *single executor* berbasis *node* tunggal.

Pada penelitian [8], membahas tentang analisa prediktif pada *big data* dengan melakukan pendekatan fungsi dari *library* pada ekosistem apache spark. Penelitian ini membuktikan dengan pemanfaatan library pada spark dapat meningkatkan performa dan efisiensi dengan berbagai komponen dan kompabilitasnya dalam mengelola data besar dibanding metode tradisional.

Berdasarkan latar belakang masalah tersebut, maka pada penelitian ini akan membahas mengenai analisa *big data* pada *cluster* komputer menggunakan komputasi terdistribusi. Peneliti akan memanfaatkan sistem paralelisasi dari sistem apache spark yang terbukti lebih cepat melakukan pemrosesan dibanding *MapReduce* sebagai media untuk melakukan *cluster* komputer yang terdistribusi serta menggunakan fungsi *library* pada spark untuk melakukan pemrosesan *big data*.

1.2 Perumusan Masalah

Perumusan masalah dari tugas akhir ini, yaitu sebagai berikut:

1. Bagaimana cara meningkatkan dan mengoptimalkan kinerja *cluster* komputer pada pemrosesan data.
2. Bagaimana merancang dan membangun sistem komputasi terdistribusi untuk pemrosesan data pada *cluster* komputer.
3. Bagaimana pengaruh spark untuk melakukan pemrosesan data pada *cluster* komputer yang terdistribusi.

1.3 Batasan Masalah

Batasan masalah dari tugas akhir ini, yaitu sebagai berikut:

1. Menggunakan Apache Spark sebagai *engine* untuk menangani data yang berskala besar dan dapat memprosesnya dengan cepat dengan memproses data secara paralel.
2. Menggunakan *Cluster* komputer mode *Distributed Computing* (Komputasi Terdistribusi) yang menawarkan skalabilitas yang tinggi untuk menjalankan aplikasi secara paralel di *cluster* komputer.
3. Menggunakan fungsi *library* yang tersedia pada apache spark yang memiliki kemampuan dan kecepatan yang sangat baik dalam melakukan pemrosesan data karena berjalan secara *in memory caching.*

1.4 Tujuan

Tujuan dari penulisan tugas akhir ini, yaitu:

1. Menerapkan dan memahami Apache Spark sebagai *engine* yang berjalan secara *Distributed Computing* pada *cluster* komputer untuk pemrosesan data.
2. Menganalisa penggunaan teknik pemrosesan komputasi yang dihasilkan pada *cluster* komputer melalui *library* apache spark.
3. Dapat mengukur hasil dari kinerja terdapat nilai RMSE, akurasi, presisi, recall, dan F1-score.

1.5 Manfaat

Manfaat dari penulisan tuags akhir ini, yaitu sebagai berikut:

1. Dapat mengetahui cara kerja pemrosesan data pada *cluster* komputer mode *Distibuted Computing* (Komputasi Terdistribusi) yang berjalan pada Apache Spark.
2. Dapat mempersingkat waktu yang diperlukan dalam melakukan pemrosesan data berskala besar yang berkerja secara paralel.

1.6 Metodologi Penelitian

Metodologi penelitian pada tugas akhir ini terdapat beberapa tahap, yaitu:

1. Studi Literatur

   Tahap awal dalam melakukan penelitian ini dengan mencari hingga mengumpulkan berbagai referensi berupa literatur terdapat pada jurnal, buku dan internet yang berkaitan tentang Analisis *Big Data* dengan Cluster Komputer menggunakan Komputasi Terdistribusi.
2. Konsultasi

   Pada tahapan melakukan konsultasi kepada dosen pembimbing serta pihak-pihak yang memiliki pemahama dan wawasan yang baik dalam mengatasi permasalahan yang di temui pada penulisan tugas akhir dengan judul Analisis *Big Data* dengan Cluster Komputer menggunakan Komputasi Terdistribusi.
3. Pembuatan Infrastruktur

   Pada tahapan ini melakukan pembuatan infrastruktur pada Apache Spark melalui *cluster* manager dengan komputasi terdistribusi.
4. Pembuatan Model

   Pada tahapan ini dilakukan perancangan pemanfaatan model *Machine Learning* menunjang untuk mengolah data.
5. Pengujian

   Pada tahapan ini melakukan pengujian terhadap proses data pada cluster yang berjalan di atas Apache Spark.
6. Analisa dan Kesimpulan

Hasil dari pengujian pada tugas akhir ini aka dianalisa baik kelebihannya serta kekurangannya dan juga menganalisa bagaimana proses *cluster* komputer secara komputasi terdistribusi pada Apache Spark dalam menangani proses data menggunakan *library* pada spark.

1.7 Sistematika Penulisan

Adapun sistematika penulisan pada tugas akhir ini agar dapat mendeskripsikan bab-bab dalam penelitian ini, antara lain:

BAB I PENDAHULUAN

Pada bab ini, berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan serta manfaat dari topik yang diangkat yaitu mengenai Analisa *Big Data* pada Cluster Komputer menggunakan Komputasi Terdistribusi.

BAB II TINJAUAN PUSTAKA

Pada bab ini, berisi dasar toeri berkaitan dengan Analisa *Big Data* pada *Cluster* Komputer menggunakan Komputasi Terdistribusi yang dikumpulkan dari berbagai sumber untuk menunjang penelitian dan penulisan tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Pada bab ini, berisi kerangka kerja dan metode yang akan dilakukan pada penelitian ini.

BAB IV HASIL DAN ANALISA

Pada bab ini, berisi analisa dan pembahasan dari hasil penelitian yang telah dilakukan untuk memproleh berbagai petunjuk yang dapat menghasilkan kesimpulan dari penelitian ini.

BAB V KESIMPULAN DAN SARAN

Pada bab ini, berisi kesimpulan dan saran dari hasil penelitian yang telah di lakukan.

DAFTAR PUSTAKA

[1] H. Kadkhodaei, A. M. Eftekhari Moghadam, and M. Dehghan, "Big data classification using heterogeneous ensemble classifiers in Apache Spark based on MapReduce paradigm," *Expert Syst. Appl.*, vol. 183, no. April, p. 115369, 2021, doi: 10.1016/j.eswa.2021.115369.

[2] H. C. Lu, F. J. Hwang, and Y. H. Huang, *Parallel and distributed architecture of genetic algorithm on Apache Hadoop and Spark*, vol. 95. Elsevier B.V., 2020. doi: 10.1016/j.asoc.2020.106497.

[3] S. Al-Saqqa, G. Al-Naymat, and A. Awajan, "A large-scale sentiment data classification for online reviews under apache spark," *Procedia Comput. Sci.*, vol. 141, pp. 183–189, 2018, doi: 10.1016/j.procs.2018.10.166.

[4] F. Abuqabita, R. Al-Omoush, and J. Alwidian, "A Comparative Study on Big Data Analytics Frameworks, Data Resources and Challenges," *Mod. Appl. Sci.*, vol. 13, no. 7, p. 1, 2019, doi: 10.5539/mas.v13n7p1.

[5] P. Dahiya and D. K. Srivastava, "Network Intrusion Detection in Big Dataset Using Spark," *Procedia Comput. Sci.*, vol. 132, pp. 253–262, 2018, doi: 10.1016/j.procs.2018.05.169.

[6] J. M. Abuin, N. Lopes, L. Ferreira, T. F. Pena, and B. Schmidt, "Big Data in metagenomics: Apache Spark vs MPI," *PLoS One*, vol. 15, no. 10 October, pp. 1–20, 2020, doi: 10.1371/journal.pone.0239741.

[7] P. Salwan* and D. V. K. Maan, "Integrating E-Governance with Big Data Analytics using Apache Spark," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 1609–1615, 2020, doi: 10.35940/ijrte.f7820.038620.

[8] M. Junaid, S. A. Wagan, N. M. F. Qureshi, C. S. Nam, and D. R. Shin, "Big data Predictive Analytics for Apache Spark using Machine Learning," in *2020 Global Conference on Wireless and Optical Technologies, GCWOT 2020*, 2020. doi: 10.1109/GCWOT49901.2020.9391620.

[9] B. Hosseini and K. Kiani, “A big data driven distributed density based hesitant fuzzy clustering using Apache spark with application to gene expression microarray,” *Eng. Appl. Artif. Intell.*, vol. 79, no. January, pp. 100–113, 2019, doi: 10.1016/j.engappai.2019.01.006.

[10] H. Elzayady, K. M. Badran, and G. I. Salama, “Sentiment Analysis on Twitter Data using Apache Spark Framework,” *Proc. - 2018 13th Int. Conf. Comput. Eng. Syst. ICCES 2018*, pp. 171–176, 2019, doi: 10.1109/ICCES.2018.8639195.

[11] T. Santosh and D. Ramesh, “Machine learning approach on apache spark for credit card fraud detection,” *Ing. des Syst. d’Information*, vol. 25, no. 1, pp. 101–106, 2020, doi: 10.18280/isi.250113.

[12] Y. K. Guntupalli, V. S. Saketh, S. Amudheswaran, and D. S. Vaishnav, “High-Scale Food Recommendation Built on Apache Spark using Alternating Least Squares,” *2020 5th IEEE Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2020 - Proceeding*, vol. 2020, 2020, doi: 10.1109/ICRAIE51050.2020.9358277.

[13] Y. Xu, H. Liu, and Z. Long, “A distributed computing framework for wind speed big data forecasting on Apache Spark,” *Sustain. Energy Technol. Assessments*, vol. 37, no. October 2019, p. 100582, 2020, doi: 10.1016/j.seta.2019.100582.

[14] A. M. Fernández, A. Troncoso, and F. M. Álvarez, “Automated Deployment of a Spark Cluster with Machine Learning Algorithm Integration,” *Big Data Res.*, vol. 19–20, p. 100135, 2020, doi: 10.1016/j.bdr.2020.100135.

[15] S. H. Liao and C. A. Yang, “Big data analytics of social network marketing and personalized recommendations,” *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–19, 2021, doi: 10.1007/s13278-021-00729-z.

[16] E. Nagy, R. Lovas, I. Pintye, Á. Hajnal, and P. Kacsuk, “Cloud-agnostic architectures for machine learning based on Apache Spark,” *Adv. Eng. Softw.*, vol. 159, no. May, 2021, doi: 10.1016/j.advengsoft.2021.103029.

[17] A. Zainab, A. Ghrayeb, H. Abu-Rub, S. S. Refaat, and O. Bouhali, “Distributed Tree-Based Machine Learning for Short-Term Load Forecasting with Apache Spark,” *IEEE Access*, vol. 9, pp. 57372–57384, 2021, doi: 10.1109/ACCESS.2021.3072609.

[18] H. Yu, “Apriori algorithm optimization based on Spark platform under big data,” *Microprocess. Microsyst.*, vol. 80, no. November 2020, p. 103528, 2021, doi: 10.1016/j.micpro.2020.103528.

[19] G. Cheng, S. Ying, B. Wang, and Y. Li, “Efficient Performance Prediction for Apache Spark,” *J. Parallel Distrib. Comput.*, vol. 149, pp. 40–51, 2021, doi: 10.1016/j.jpdc.2020.10.010.

[20] F. Yang, H. Wang, and J. Fu, “Improvement of recommendation algorithm based on Collaborative Deep Learning and its Parallelization on Spark,” *J. Parallel Distrib. Comput.*, vol. 148, pp. 58–68, 2021, doi: 10.1016/j.jpdc.2020.09.014.

[21] A. A. Hagar and B. W. Gawali, “Apache Spark and Deep Learning Models for High-Performance Network Intrusion Detection Using CSE-CIC-IDS2018,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/3131153.

[22] O. Azeroual and A. Nikiforova, “Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data,” *Inf.*, vol. 13, no. 2, 2022, doi: 10.3390/info13020058.

[23] T. Singh, S. Gupta, M. Kumar, and M. Kumar, “Performance Analysis and Deployment of Partitioning Strategies in Performance Analysis and Apache Deployment of Partitioning Strategies in Spark Apache Spark,” *Procedia Comput. Sci.*, vol. 218, no. 2022, pp. 594–603, 2023, doi: 10.1016/j.procs.2023.01.041.

[24] R. A. Ngowi, “Full Impact of Big Data in the Supply Chain Directorate of Research , Publication and Postgraduate Studies Dar -es -Salaam Campus College Masters of Business Administration in Corporate Management,” no. June, 2021.

[25] D. Sitaram and G. Manjunath, *Related Technologies*. 2012. doi: 10.1016/b978-1-59749-725-1.00009-3.

[26] P. Singh and S. Singh, "ORIGINAL RESEARCH A data structure perspective to the RDD-based Apriori algorithm on Spark," *Int. J. Inf. Technol.*, 2019, doi: 10.1007/s41870-019-00337-3.

[27] A. Spark, "Unified engine for large-scale data analytics," 2023. https://spark.apache.org/ (accessed Feb. 16, 2023).

[28] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12343 LNCS, pp. 503–515, 2020, doi: 10.1007/978-3-030-62008-0_35.

[29] C. Gardner and D. C. T. Lo, "PCA embedded random forest," *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2021-March, pp. 1–6, 2021, doi: 10.1109/SoutheastCon45413.2021.9401949.

[30] S. M. Taghavinejad, "Intrusion Detection in IoT-Based Smart Grid Using Hybrid Decision Tree," *2020 6th Int. Conf. Web Res. ICWR 2020*, 2020.

[31] F. J. Yang, "An extended idea about decision trees," *Proc. - 6th Annu. Conf. Comput. Sci. Comput. Intell. CSCI 2019*, pp. 349–354, 2019, doi: 10.1109/CSCI49370.2019.00068.

[32] L. Li, Y. Zhang, W. Chen, S. K. Bose, M. Zukerman, and G. Shen, "Naïve Bayes classifier-assisted least loaded routing for circuit-switched networks," *IEEE Access*, vol. 7, pp. 11854–11867, 2019, doi: 10.1109/ACCESS.2019.2892063.

[33] I. A. P. Banlawe, J. C. D. Cruz, J. C. P. Gaspar, and E. J. I. Gutierrez, "Optimal Frequency Characterization of Mango Pulp Weevil Mating Activity using Naïve Bayes Classifier Algorithm," *Proceeding - 2021 IEEE 17th Int. Colloq. Signal Process. Its Appl. CSPA 2021*, no. March, pp. 116–120, 2021, doi: 10.1109/CSPA52141.2021.9377277.

[34] D. Lei, J. Tang, Z. Li, and Y. Wu, “Using low-rank approximations to speed up kernel logistic regression algorithm,” *IEEE Access*, vol. 7, pp. 84242–84252, 2019, doi: 10.1109/ACCESS.2019.2924542.

[35] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Raheem, “Performance comparison of support vector machine , random forest , and extreme learning machine for intrusion detection,” vol. 3536, no. c, pp. 1–7, 2018, doi: 10.1109/ACCESS.2018.2841987.

[36] C. B. E. N. Issaid and A. Member, “User Clustering for MIMO NOMA via Classifier Chains and Gradient-Boosting Decision Trees,” pp. 211411–211421, 2020, doi: 10.1109/ACCESS.2020.3038490.