SKRIPSI

STUDI DAN IMPLEMENTASI POLA DESAIN MAPREDUCE II

Irvan Wijaya Ludianto

NPM: 2015730008

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019

UNDERGRADUATE THESIS

STUDY AND IMPLEMENTASION MAPREDUCE DESIGN PATTERN II

Irvan Wijaya Ludianto

NPM: 2015730008

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019

LEMBAR PENGESAHAN

STUDI DAN IMPLEMENTASI POLA DESAIN MAPREDUCE II

Irvan Wijaya Ludianto

NPM: 2015730008

Bandung, 24 Juli 2019

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji	Anggota Tim Penguji
Raymond Chandra Putra, S.T., M.T.	Dr.rer.nat. Cecilia Esti Nugraheni

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 POLA DESAIN MAPREDUCE II

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 24 Juli 2019

Meterai
Rp. 6000

Irvan Wijaya Ludianto
NPM: 2015730008

ABSTRAK

Istilah *Big Data* sudah bukanlah lagi istilah yang asing. Istilah ini muncul karena jumlah data yang terus bertambah dan semakin besar setiap tahunnya. Data ini tentu saja perlu diolah untuk menghasilkan suatu informasi dan menarik suatu kesimpulan, namun hal ini sulit dilakukan dan membutuhkan waktu yang lama mengingat ukuran data yang begitu besar. Oleh karena itu sebuah *framework* bernama *Hadoop* diciptakan agar dapat mengolah *Big Data* secara terdistribusi dengan menggunakan model pemrograman *MapReduce*.

MapReduce terus digunakan dan menjadi semakin populer hingga akhirnya terlihatlah banyak permasalahan umum yang membutuhkan solusi yang serupa dalam mengolah data dengan *MapReduce*. Masalah-masalah umum tersebut dapat diselesaikan dengan menggunakan *design pattern* sehingga lebih mudah untuk dipecahkan dan digunakan kembali. Jenis-jenis *design pattern* antara lain adalah *summarization*, *filtering*, *data organization*, *join*, *meta*, dan *input and output*. Tentu saja pola desain tersebut harus dimodifikasi, dipelajari, dan diuji performanya sebelum digunakan.

Penelitian ini memuat 3 macam pola desain yang dikostumisasi sehingga dapat digunakan untuk beberapa kasus nyata tertentu yang menggunakan data berbentuk *JSON*. Ketiga pola desain yang dibuat tersebut adalah *Filter Pattern*, *Join Pattern*, dan *Input and Output Pattern*. Tentu ketiga pola desain ini memiliki kebutuhan dan fungsi yang berbeda-beda. Seluruh perangkat lunak demo untuk pola-pola desain tersebut telah berhasil disusun dan dapat digunakan sesuai dengan kasus-kasus nyata tertentu dimana data yang digunakan memiliki format *JSON*. Seluruh pola desain yang dibuat kemudian diuji dan dilihat waktu eksekusinya dengan menggunakan beberapa data dengan ukuran yang berbeda.

Hasil pengujian membuktikan bahwa performa yang dimiliki oleh setiap pola desain berbeda-beda tergantung dari komponen yang digunakan dan implementasi dari pola-pola desain tersebut. Selain itu performa itu juga dipengaruhi oleh berbagai faktor lainnya seperti ukuran dari data yang diproses, jumlah *Block* data, performa dari komputer yang melakukan proses sendiri, dan hal-hal lainnya.

Kata-kata kunci: *Big Data*, *Hadoop*, *MapReduce*, Pola Desain, Pola *Filter*, Pola *Join*, Pola *Input and Output*

ABSTRACT

The term of Big Data is no longer a foreign term. This term arises because the amount of data is keep increasing every year. The increasing data of course needs to be processed to produce an information and draw a conclusion, but this is hard to do and requires a long processing time considering the size of the data is so large. Therefore a framework called Hadoop was created so that it can process Big Data in a distributed manner using the MapReduce programming model.

MapReduce continues to be used and is becoming increasingly popular until finally there are many common problems that require similar solutions in processing data with MapReduce. These general problems can be solved using a design pattern so that it is easier to solve and reuse. The types of pattern design include summarization, filtering, data organization, join, meta, and input and output. Of course these design patterns must be modified, studied, and tested for performance before use.

This study contains 3 kinds of customized design patterns so that it can be used for certain specific cases that use JSON formatted data. The three design patterns created are Filters Pattern , Join Patterns, and Input and Output Patterns. Of course these three design patterns have different needs and functions. All demo software for these design patterns have been successfully build and can be used in accordance with certain real cases where the input data used has the JSON format. All design patterns created are then tested and seen when they are executed by using several data of different sizes.

The test results prove that the performance of each design pattern varies depending on the components used and the implementation of the design patterns. In addition, performance is also influenced by various other factors such as the size of the data being processed, the number of blocks of data, the performance of the computer that performs the process itself, and several other things.

Keywords: *Big Data*, *Hadoop*, *MapReduce*, *Design Pattern*, *Filter Pattern*, *Join Pattern*, *Input and Output Pattern*

Skripsi ini dipersembahkan kepada Ayah, Ibu, saudara-saudara, teman-teman, dan seluruh orang yang membutuhkan hasil penelitian dari penelitian ini.

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis bisa terus diberikan kekuatan sehingga dapat menyelesaikan skripsi ini sesuai dengan waktu yang diharapkan. Penulis menyadari bahwa di dalam skripsi ini masih terdapat banyak kekurangan yang disebabkan oleh keterbatasan dari segi kemampuan dan pengetahuan yang dimiliki oleh penulis. Oleh karena itu penulis dengan senang hati untuk menerima saran dan kritik yang membangun.
Pada kesempatan kali ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Orang tua penulis yaitu Lo Tjauw Fong dan Tan Kim Hwei yang selalu memberikan dukungan sehingga penulis dapat terus memiliki kekuatan untuk terus mengerjakan dalam keadaan sesulit apapun.

2. Ibu Veronica Sri Moertini sebagai dosen pembimbing yang selalu memberikan arahan yang baik berupa saran yang kritis sehingga skripsi ini dapat diselesaikan dengan hasil yang baik.

3. Bapak Raymond Chandra Putra dan Ibu Cecilia Esti Nugraheni sebagai dosen penguji yang telah memberikan masukan yang kritis sehingga hasil dari skripsi ini semakin layak dan baik.

4. Para admin lab yang bekerja dengan sangat baik dan giat sehingga setiap kali penulis mengalami masalah komputer saat melakukan pengujian, masalah tersebut dapat diselesaikan dengan cepat.

5. Anggota Founder Nibble Softworks yang dapat mengerti keadaan penulis dan memberikan waktu bagi penulis untuk menyusun skripsi sehingga skripsi dapat diselesaikan tepat pada waktunya.

6. Teman-teman terdekat penulis yang diantaranya adalah anggota *party DOTA* (Matthew, Edrick, Cornel, Himawan, dll), anggota *B&B* (Bernard, Raymond, Nico, dll),dan anggota grup *Terong*(Melinda dan Joshua) yang saling memberikan dukungan dan saling tolong menolong dalam berbagai hal.

7. Seluruh rekan-rekan lainnya yang berada satu jurusan maupun diluar jurusan yang tidak dapat disebutkan satu persatu.

Akhirnya penulis berharap semoga amal baik dari semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini mendapatkan balasan pahala dari Tuhan Yang Maha Esa. Semoga apa yang telah ditulis dalam skripsi ini dapat bermanfaat bagi penulis dan seluruh pihak yang membutuhkan hasil penelitian ini.

Bandung, Juli 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Data adalah catatan atas fakta yang telah terjadi. Dengan adanya kumpulan data kita dapat menyimpulkan sesuatu bila kumpulan data tersebut memiliki suatu hubungan satu sama lain serta memiliki ukuran yang cukup besar. Hal tersebut membuat orang-orang berlomba-lomba untuk mendapatkan data agar mereka dapat melihat suatu informasi yang berharga. Ukuran data terus bertambah setiap tahunnya dan pada akhirnya berukuran sangat besar yang sering kita dengar dengan sebutan *Big Data*. Ukuran dari *Big Data* cukup relatif, ada orang yang mengatakan data berukuran ratusan *Gigabyte* adalah *Big Data* namun ada juga orang yang mengatakan bahwa data berukuran *Terabyte* baru termasuk *Big Data*. Oleh karena itu secara umum *Big Data* adalah kumpulan data yang berukuran sangat besar sehingga perlu waktu yang lama untuk diproses lebih lanjut sehingga dapat menjadi informasi yang berharga.

Untuk menganalisa data dengan ukuran yang sangat besar tidaklah mudah serta memakan banyak waktu dan biaya. Bila waktu yang dibutuhkan untuk mengubah suatu data menjadi informasi terlalu lama maka informasi yang didapatkan terkadang sudah tidak lagi berharga. Hal tersebut menghasilkan kebutuhan komputasi yang cepat. Komputasi paralel adalah sebuah solusi dimana proses dapat dilakukan secara bersamaan oleh banyak sistem sehingga pemrosesan *Big Data* dapat dilakukan dengan lebih cepat. Kecepatan adalah salah satu keuntungan komputasi paralel namun berbagai masalah yang harus ditangani oleh komputasi paralel juga harus diperhitungkan.

MapReduce adalah salah satu teknik yang pertama kali diperkenalkan oleh *Google* [1] untuk melakukan komputasi pada data yang berskala sangat besar secara terdistribusi dan paralel. Teknik ini secara umum akan memecah suatu masukan(*input*) kedalam sebuah daftar (*map*) yang berisikan pasangan kunci dan nilai. Daftar tersebut yang kemudian akan diproses kembali (*reduce*) untuk menghasilkan keluaran(*output*) yang diinginkan. *MapReduce* menjadi suatu solusi yang sangat baik untuk melakukan pemrosesan *Big Data* sehingga pada akhirnya muncullah sebuah *framework* yang bernama *Hadoop*. *Hadoop* adalah *framework* atau *platform open source* berbasis Java yang menggunakan teknologi *Google MapReduce* serta *Google File System (GFS)* sebagai fondasinya. Pada skripsi ini *framework Hadoop* inilah yang akan digunakan.

Setelah *MapReduce* menjadi populer dan sering digunakan akhirnya orang-orang tersadar bahwa banyak permasalahan umum yang selalu diselesaikan dengan solusi serupa yang dibuat berkali-kali. Masalah tersebut memacu munculnya berbagai pola desain. Ada 6 pola desain pada buku "Design Pattern Mapreduce" [2] yang diantaranya adalah *summarization*, *filtering*, *data organization*, *join*, *meta*, dan *input and output*. Pola-pola desain ini memiliki banyak keuntungan namun tentu saja suatu permasalahan tidak dapat diselesaikan hanya dengan menjiplak suatu pola secara mentah-mentah dan perlu modifikasi sesuai kebutuhan [2]. Hal tersebut memacu kebutuhan untuk mempelajari, mengkostumisasi, serta mengukur performa dari setiap pola desain. Skripsi ini meneliti tentang 3 tipe pola desain, yaitu *Filtering Pattern*, *Join Pattern*, dan *Input and Output Pattern* dari segi konsep yang digunakan serta kinerja yang dihasilkan dari ketiga pola tersebut. Untuk dapat meneliti hal-hal tersebut maka perlu dibuat perangkat lunak yang generik untuk setiap pola sehingga dapat diuji untuk berbagai kasus-kasus nyata tertentu.

1.2 Rumusan Masalah

Berdasarkan apa yang sudah dijelaskan pada latar belakang subbab 1.1 maka berikut adalah rumusan masalah yang akan dibahas pada skripsi ini:

1. Bagaimana menerapkan pola desain *Filtering*, *Join*, dan *Input and Output* pada kasus-kasus nyata tertentu?
2. Bagaimana performa dari pola desain *Filtering*, *Join*, dan *Input and Output*?
3. Bagaimana membangun perangkat lunak demo untuk pola desain *Filtering*, *Join*, dan *Input and Output*?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disusun pada subbab 1.2 maka tujuan dari penelitian ini adalah:

1. Membangun modul-modul program untuk pola desain *Filtering*, *Join*, dan *Input and Output*.
2. Melakukan eksperimen untuk mengukur kinerja dari pola desain *Filtering*, *Join*, dan *Input and Output*.
3. Membangun perangkat lunak demo yang dapat menjalankan pola desain *Filtering*, *Join*, dan *Input and Output* untuk kasus-kasus tertentu.

1.4 Batasan Masalah

Batasan masalah pada skripsi ini antara lain adalah:

1. Skripsi ini hanya akan memuat 3 jenis pola desain *MapReduce* yang ada, yaitu pola desain *Filtering*, *Join*, dan *Input and Output*.
2. Tidak menangani keamanan atau validasi pengguna pada perangkat lunak demo.

1.5 Metodologi

Metodologi yang digunakan dalam pembuatan skripsi ini adalah:

1. Melakukan studi literatur tentang *Hadoop* secara keseluruhan yang meliputi *library Hadoop* sendiri, *HDFS* (Hadoop Distributed File System), dan *MapReduce*.
2. Melakukan studi literatur tentang pola desain *Filtering*, *Join*, dan *Input and Output* pada *MapReduce*.
3. Mengimplementasikan program *word count* sebagai eksperimen agar lebih mengerti dan terbiasa dengan konsep *MapReduce* dan *Hadoop*.
4. Mengimplementasikan 12 kode program *design pattern MapReduce* yang ada pada buku "Design Pattern Mapreduce" [2] sebagai eksperimen agar lebih mengerti tentang konsep *design pattern MapReduce*.
5. Mengkonfigurasi dan mencoba menjalankan program *MapReduce* pada mode *Pseudo-distributed* dan mode *Fully-distributed*.

6. Mencari *data set* yang dapat digunakan untuk pola desain *Filtering*, *Join*, dan *Input and Output*.
7. Merancang modul-modul program untuk pola desain *Filtering*, *Join*, dan *Input and Output*.
8. Mengimplementasikan modul-modul program yang sudah dirancang.
9. Menguji tingkat kebenaran dan kinerja dari masing-masing pola desain.
10. Membuat dokumen skripsi.

1.6 Sistematika Pembahasan

1. Bab 1 Pendahuluan
   Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, serta sistematika pembahasan yang digunakan dalam penelitian serta penyusunan dokumen skripsi ini.
2. Bab 2 Landasan Teori
   Bab ini membahas mengenai teori-teori yang digunakan dalam penyusunan skripsi ini. Teori-teori tersebut antara lain adalah teori mengenai *Big Data*, sistem terdistribusi *Hadoop*, *HDFS* (*Hadoop Distributed File System*), *MapReduce*, pola desain pada umumnya, pola desain *Filtering*, pola desain *Join*, dan pola desain *Input and Output*.
3. Bab 3 Studi Eksplorasi Hadoop
   Bab ini membahas mengenai studi eksplorasi menggunakan *Hadoop* berupa eksperimen kecil yang dilakukan. Eksperimen tersebut memuat program *Word Count* serta 12 *design pattern* yang berasal dari buku "Design Pattern Mapreduce" [2].
4. Bab 4 Analisis dan Perancangan
   Bab ini membahas mengenai analisis serta perancangan perangkat lunak yang dibangun pada skripsi ini. Analisis dan perancangan yang dimuat adalah kebutuhan dari perangkat lunak yang ingin disusus, analisis masukan dan keluaran yang diharapkan, perancangan *GUI*(*Graphical User Interface*), perancangan kelas pada perangkat lunak.
5. Bab 5 Implementasi dan Pengujian
   Bab ini berisi mengenai speseifikasi perangkat lunak yang digunakan selama pengujian, hasil implementasi perangkat lunak, hasil implementasi antarmuka, pengujian kebenaran dari modul-modul program hasil implementasi, eksperimen pengukuran performa yang dilakukan untuk mengukur kinerja dari tiap pola desain yang telah diimplementasikan, dan kesimpulan hasil eksperimen.
6. Bab 6 Kesimpulan dan Saran
   Bab ini membahas mengenai kesimpulan-kesimpulan dan saran-saran dari penulis yang berhasil didapatkan dari penelitian ini.