**PERBANDINGAN PERFOMANSI PENGOLAHAN DATA MENGGUNAKAN *HADOOP MAPREDUCE* DAN *DBMS***



**PROPOSAL TUGAS AKHIR**

|  |  |  |
| --- | --- | --- |
| NPM  Nama  Konsentrasi | :  :  : | 0615101038  MUHAMMAD FADILLAH  *APPLIED DATABASE* |

******

**SK.Ketua Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT)**

**Nomor : 041/BAN-PT/AK-XIV/S1/XII/2011**

**TEKNIK INFORMATIKA – FAKULTAS TEKNIK**

**UNIVERSITAS WIDYATAMA**

**BANDUNG**

**2018**

**REVIEW PROPOSAL PENELITIAN**

NPM : 0615101038

Nama : Muhammad Fadillah

Konsentrasi : *Applied Database*

Judul : Perbandingan Perfomansi Pengolahan Data

Menggunakan Hadoop Mapreduce dan DBMS.

|  |  |
| --- | --- |
| Aspek Penilaian | Hasil Review |
| Originalitas |  |
| Kelegkapan Informasi |  |
| Kelayakan Ilmu dan Waktu Pelaksanaan |  |

Kesimpulan :

|  |
| --- |
| Diterima/ Ditolak/ Diperbaiki  Alasan : |

Bandung, 18 Desember 2018

|  |  |
| --- | --- |
| Menyetujui  Ka. Lab. *Applied Database*  **Ardhian Ekawijana, S.T., M.T** | Mengetahui  Sek. Prodi Teknik Informatika  **Ari Purno Wahyu, S.Kom, M.Kom.** |

# BAB I PENDAHULUAN

## LATAR BELAKANG MASALAH

Dalam 20 tahun terakhir, jumlah data telah meningkat dalam skala besar pada berbagai bidang. Menurut *International Data Corporation (IDC)* pada tahun 2011, jumlah data yang dibuat diseluruh dunia berjumlah sekitar 1,8 zettabytes [1], pada 2013 meningkat menjadi 4,4 zettabytes dan diprediksi meningkat sepuluh kali lipat menjadi 44 zettabytes pada 2020 [2]. Fenomena *big data* bukan hanya berbicara tentang kemampuan untuk menyimpan dan mengelola data yang berukuran sangat besar dan bervariatif, tetapi juga bagaimana untuk mengelola data tersebut.

Sejalan dengan hal tersebut, maka pengolahan terhadap *big data* merupakan suatu hal yang kritis pula. Pengolahan terhadap *big data* bukan merupakan suatu hal yang mudah. Pengolahan *big data* tidak dapat disamakan dengan pengolahan data dengan ukuran yang relatif kecil. *Single computer* akan terhambat kinerjanya atau juga tidak akan dapat mengolah data jika ukurannya melebihi kapasitas *memory* pada komputer tersebut. Oleh karena itu diperlukanlah suatu *tool* atau kerangka kerja yang dapat membantu proses pengolahan terhadap *big data,* salah satunya yaitu *Apache Hadoop*. *Apache Hadoop* sendiri merupakan kerangka kerja yang dapat diimplementasikan pada *single computer* ataupun *multiple computer* dalam suatu jaringan tertentu.

Mengetahui pentingnya *issue* mengenai pengolahan *big data* sekarang ini, maka eksplorasi terhadap *Apache Hadoop* dirasa cukup penting. Eksplorasi ini adalah sebagai media implementasi atau praktik langsung agar lebih memahami cara kerja serta fitur-fitur yang ditawarkan *Apache Hadoop* sehingga dapat meningkatkan kinerja komputer dalam mengolah *big data* bila dibanding dengan pemrosesan dengan *single computer*. Terlebih lagi, eksplorasi ini diharapkan dapat mengubah paradigma dalam pengolahan *big data* yang terlalu terpusat pada *single computer* dan SQL (*Structured Query Language*) menuju pada penggunaan komputer terdistribusi. Berdasarkan pemaparan diatas penulis akan menganalisis kedua metode tersebut agar dapat menentukan metode mana yang lebih baik. Proses analisis kedua metode eksplorasi ini menggunakan data peminjaman buku perpustakaan Universitas Widyatama.

## RUMUSAN MASALAH

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang masalah, maka dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana mengimplementasikan teknologi Hadoop MapReduce pada pengolahan data peminjaman perpustakaan Universitas Widyatama ?
2. Bagaimana cara mengimplementasikan metode MapReduce berbasis HDFS (Hadoop Distributed File System) pada pengolahan data peminjaman perpustakaan Universitas Widyatama ?
3. Bagaimana performansi pengolahan data menggunakan metode MapReduce dibandingkan pengolahan data menggunakan metode DBMS?

## TUJUAN

Penelitian pada permasalahan yang telah dijelaskan pada bagian latar belakang masalah, bertujuan sebagai berikut :

1. Mengimplementasikan teknologi Hadoop MapReduce pada pengolahan data peminjaman perpustakaan Universitas Widyatama.
2. Mengimplementasikan metode MapReduce berbasis HDFS (Hadoop Distributed File System) pada pengolahan data peminjaman perpustakaan Universitas Widyatama.
3. Menguji dan menganalisa performa pengolahan data menggunakan metode MapReduce dibandingkan pengolahan data menggunakan metode DBMS.

## BATASAN MASALAH

Penelitian ini hanya untuk mengimplementasikan teknologi Hadoop MapReduce dan membandingkan perfomansi dengan teknologi DBMS pada pengolahan data peminjaman perpustakaan Universitas Widyatama dan dibatasi dengan hal-hal sebagai berikut:

1. Perancangan sistem yang akan dibangun adalah *single node.*
2. Sistem yang akan dibangun menggunakan Hadoop Distributed File System sebagai sistem file terdistribusi.
3. Platform yang digunakan yaitu *Apache Hadoop* dan *SQL Server*.
4. Perfomansi pengolahan data yang akan dibahas terbatas pada faktor waktu eksekusi, *CPU usage*, dan *memory usage* yang terjadi pada pengujian sistem.

## METODOLOGI

Metodologi penelitian yang diterapkan dalam penulisan tugas akhir ini meliputi:

* 1. **Identifikasi masalah**

1. Bagaimana mengimplementasikan teknologi Hadoop MapReduce pada pengolahan data peminjaman perpustakaan Universitas Widyatama.
2. Bagaimana cara mengimplementasikan metode MapReduce berbasis HDFS (Hadoop Distributed File System) pada pengolahan data peminjaman perpustakaan Universitas Widyatama.
3. Bagaimana performansi pengolahan data menggunakan metode MapReduce dibandingkan pengolahan data menggunakan metode DBMS.
   1. **Studi Literatur**

Melakukan pencarian referensi teori-teori kepustakaan yang berupa *paper*, jurnal, dan buku-buku untuk menunjang penyusunan laporan tugas akhir.

* 1. **Analisis dan Perancangan**

Dalam penelitian ini membuat sistem pengolahan data peminjaman perpustakaan Universitas Widyatama menggunakan Hadoop MapReduce, dibutuhkan langkah-langkah perencanaan sebelum *Hadoop* dapat benar-benar diimplementasikan. Langkah-langkah tersebut dilakukan untuk mempersiapkan program pengolah data dengan *Hadoop* dapat dijalankan, dapat menghasilkan suatu keluaran serta dapat dipahami cara kerjanya. Langkah-langkah tersebut adalah sebagai berikut:

1. Melakukan instalasi dan konfigurasi *Hadoop* pada *namenode*, dan *datanode*.
2. Mengolah data dari perpustakaan menjadi bentuk yang lebih mudah untuk diolah lebih lanjut dalam program *Hadoop*.
3. Menambah ukuran data peminjaman perpustakaan menjadi berukuran cukup besar sekitar 9,6 GB (*Gigabytes*) karena dirasa ukuran data tersebut masih belum terlalu besar serta menginputkannya kedalam HDFS.
4. Membuat 5 buah program dengan permasalahan yang berbeda untuk mengolah data peminjaman perpustakaan berdasarkan paradigma *mapreduce*.

Setelah langkah-langkah tersebut dilakukan, maka program *Mapreduce* dengan *Hadoop* baru dapat dijalankan sehingga memberikan suatu hasil analisa dari sekumpulan data berukuran besar.

* 1. **Implementasi**

Pada tahap implementasi, sesuai dari analisis dan perancangan yang telah ditentukan, menghasilkan sebuah sistem pengolahan data peminjaman perpustakaan Universitas Widyatama menggunakan Hadoop MapReduce.

* 1. **Kesimpulan**

Membuat kesimpulan berdasarkan data yang didapat dari hasil implementasi.

**BAB II  
LANDASAN TEORI**

## LANDASAN TEORI

Berdasarkan latar belakang dan rumusan masalah berikut adalah landasan teori yang dibutuhkan saat penelitian :

### Big Data

*Big data* merupakan konsep data yang memiliki dimensi 3Vs volume, variasi, dan kecepatan data. Volume merujuk pada banyaknya data, variasi merujuk pada jumlah dari tipe data yang digunakan dan kecepatan merujuk pada kecepatan dari pemrosesan data. Berdasarkan model 3Vs, tantangan yang dihadapi untuk mengelola *big data* adalah dari ketiga dimensi tersebut, bukan hanya mengenai jumlah atau volume data saja. Doug Laney analis dari *Gartner* memperkenalkan konsep 3Vs pada saat publikasi riset MetaGroup yang bertajuk *3D Data Management: Controling data volume, variety, and velocity* [5].

### Distributed Systems (DS)

DS adalah adalah suatu sistem dimana komponen-komponennya terletak pada suatu jaringan komputer, berkomunikasi dan mengkoordinasikan tindakan mereka dengan hanya mengirim pesan. Definisi tersebut mengikuti karakteristik utama dari DS yaitu komponen yang digunakan bersamaan, kurangnya waktu global, dan kegagalan komponen yang bersifat independen[4].

### Apache Hadoop

*Hadoop* merupakan sebuah *platform* yang menyediakan penyimpanan terdistribusi dan kemampuan komputasi. *Hadoop* pertama kali disusun untuk memecahkan masalah skalabilitas yang ada pada Nutch (sebuah *open source* dan *search engine*). Sebelumnya *Google* telah mempublikasikan tulisan mengenai *distributed file system*, *Google File System* (GFS), dan *MapReduce*, yang merupakan sebuah kerangka kerja komputasi untuk pemrosesan paralel. Keberhasilan dari implementasi konsep tersebut pada Nutch menghasilkan dua buah proyek yang terpisah, proyek kedua tersebut adalah menjadi *Hadoop*.

Secara lebih tepat, seperti terlihat pada Gambar 5.1, *Hadoop* merupakan *master-slave architecture* yang terdiri dari *Hadoop Distributed File System* (HDFS) untuk media penyimpanan serta *MapReduce* untuk kemampuan komputasi. Sifat intrinsik dari *Hadoop* adalah partisi data dan komputasi paralel dari sumber data yang besar. Media penyimpanan tersebut dan skala kemampuan komputasi dengan penambahan *hosts* untuk *Hadoop cluster,* serta dapat mencapai ukuranvolume hingga *petabytes* pada *clusters* dengan ribuan *hosts* [3].



**Gambar 5.1 Arsitektur Hadoop [3]**

### Hadoop Distributed File System (HDFS)

HDFS adalah komponen penyimpanan Hadoop. HDFS adalah file system terdistribusi yang dimodelkan setelah tulisan mengenai Google File System (GFS). HDFS dioptimasi untuk throughput yang tinggi dan berkerja paling baik ketika membaca serta menulis file besar (gigabytes atau lebih besar). Pengaruh HDFS ini adalah untuk mendukung ukuran blok yang besar (untuk file system) dan optimasi data lokal untuk mengurangi input/output jaringan [3].

HDFS memiliki dua jenis node yang beroperasi dalam arsitektur master-slave. Sebuah namenode (master) dan beberapa datanodes (slaves). Namenode mengelola filesystem namespace. Namenode menjaga filesystem tree dan metadata untuk semua file dan direktori dalam tree. Datanode merupakan bagian yang berkerja keras dalam fileysystem. Datanode menyimpan dan mendapatkan blok ketika diperintahkan oleh (aplikasi client dan namenode), dan datanode juga melaporkan balik kepada namenode secara periodik dengan daftar blok yang disimpan dalam datanode tersebut [7].

### Hadoop Mapreduce

*MapReduce* adalah sebuah model pemrograman untuk pemrosesan data. Model ini cukup sederhana, tetapi tidak terlalu sederhana dalam mengekspresikan suatu program tertentu. Lebih penting, program *MapReduce* secara inheren bersifat paralel, sehingga menempatkan analisis data skala besar kepada siapapun dengan cukup mesin sebagai penyelesaiannya. *MapReduce* pun hadir sebagai solusi untuk mengolah data yang besar [7].

Model *MapReduce* menyederhanakan proses paralel dengan meringkas kompleksitas yang terlibat dalam berkerja dengan *distributed system* seperti komputasi paralel, pembagian kerja, serta berurusan dengan *hardware* dan *software* yang tidak dapat dihandalkan. Dengan abstraksi ini, *MapReduce* memperbolehkan programmer untuk fokus pada kebutuhan bisnis, daripada berurusan dengan komplikasi *distributed system*. Untuk model kerja dari algoritma *mapreduce* dapat dilihat pada Gambar 5.2 [3].

**** Gambar 5.2 Penyerahan *job* dari *client* ke *Mapreduce* [3]**

Operasi *map* adalah dengan membagi *input file* secara paralel menjadi beberapa bagian yang dinamakan *filesplits*. Jika sebuah *file* tunggal terlalu besar maka akan mempengaruhi waktu membaca sehingga *file* tersebut dibagi menjadi beberapa bagian. Proses pembagian *file* tidak mengerti apapun mengenai struktur logika dari *input file*, sebagai contoh *file* berbasis teks bergaris dibagi kedalam beberapa bagian dan dipecah secara *byte*. Lalu *map task* akan dibuat per *filesplit* yang ada.

Ketika sebuah *reduce task* dijalankan, proses *reduce* tersebut mengambil input yang terbagi kedalam beberapa *file* dalam semua *nodes* yang digunakan saat menjalankan *map tasks*. Setelah semua data yang tersedia secara lokal ditambahkan kedalam satu *file* pada fase penambahan. *File* tersebut kemudian digabung dan diurutkan sehingga pasangan *key value* untuk *key* tertentu akan bersebelahan. Hal ini membuat operasi *reduce* yang sebenarnya menjadi sederhana: *file* dibaca secara berurutan dan *value* dikirim ke proses *reduce* secara berulang-ulang berdasarkan *key* yang ditemui [6].

### Java

Java, yang paling diingat adalah bahwa Java merupakan bahasa pertama yang memperbolehkan *developers* untuk membuat *cross-platform networked software* untuk diterapkan di *internet.* Sekali menggunakan Java, seseorang dapat membuang gagasan bahwa semua *software* harus dibangun dan dijalankan pada suatu *platform* yang sama, kemudian dilakukan *porting* bila program tersebut butuh untuk diterapkan pada sistem lainnya [8].

### Structured Query Language(SQL)

Menurut raharjo (2011:55), SQL yaitu kependekan dari Structured Query Language, yang merupakan bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan database.

Menurut ichwan (2011:20), kegunaan bahasa SQL yaitu: (a) membangun basis data, (b) menjalankan query terhadap basis data, (c) melakukan penambahan, pengurangan, perubahan terhadap data yang ada.

### Database Management System (DBMS)

Sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan menyediakan akses ke dalam basis data (Connoly, 2002, p16). Pada dasarnya, sebuah DBMS menyediakan fasilitas sebagai berikut:

1. Data Definition Language (DDL)

DDL adalah bahasa dalam DBMS yang digunakan untuk mendefinisikan tipe data, struktur data, dan batasan sebuah data yang disimpan didalam basis data (Connoly, 2002, p16).

1. Data Manipulation Language (DML)

DML adalah bahasa dalam DBMS yang digunakan pengguna untuk melakukan modifikasi dan pengambilan data pada suatu basis data.Modifikasi data terdiri dari: penambahan (insert), pembaruan (update), penghapusan (delete), seleksi data (selection) dan membuat tabel maya (view) (Connoly, 2002, p16).

1. Pengendalian Akses Ke Dalam Basis Data

Fasilitas pengendalian akses ke dalam basis data sebagai berikut:

* 1. Sistem keamanan, mencegah pengguna yang tidak memiliki akses kedalam basis data untuk mengakses basis dataintegritas.
  2. Sistem integritas, mempertahankan konsistensi data yang disimpan.
  3. Pengendalian share data, mengizinkan berbagi akses didalam basis data.
  4. Sistem pengendalian recovery, mengembalikan data ke kondisi sebelum terjadi kegagalan sistem baik perangkat keras maupun lunak pada basis data.
  5. Katalog deskripsi data bagi akses pengguna, terdiri dari deskripsi data yang disimpan dalam basis data

1. Mekanisme View

Mekanisme view berfungsi menampilkan data yang hanya diperlukan saja oleh pengguna.

# JADWAL PENELITIAN TUGAS AKHIR

## JADWAL PENELITIAN TUGAS AKHIR

**Tabel 7.1 Jadwal Penelitian Tugas Akhir**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO | Kegiatan | Minggu Ke- | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1 | Studi Literatur dan Pengumpulan data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Proposal Tugas Akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Analisis dan Perancangan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Pengujian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Sidang |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Pembuatan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## USULAN PEMBIMBING

* Bapak Ardhian Ekawijana, S.T., M.T.
* Bapak Ardiles Sinaga, S.T, M.T.

## DAFTAR PUSTAKA

1. M. Chen, S. Mao, and Y. Liu, “Big Data : A Survey,” *Mob. Networks Appl.*, vol. 19, no. 2, pp. 171–209, 2014.
2. T. White, *Hadoop: The Definitive Guide*, 4th ed. O’Reilly, 2015.
3. Holmes, A. 2012. Hadoop in Practice. New York: Manning Publications Co.
4. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. 2012. Distributed Systems: Concepts and Design 5th edition. Pearson Education.
5. Rouse,M.2013.Big Data.URI = http://whatis.techtarget.com/definition/3Vs.
6. Taggart, A. 2011. How Map and Reduce operations are actually carried out. URI = http://wiki.apache.org/hadoop/HadoopMapReduce.
7. White, T. 2012. Hadoop: The Definitive Guide (3rd ed.). O’Reilly Media, Inc.
8. Potts, A., & Friedel, J. D. 1996. Java Programming Language Handbook. Scottsdale: Keith Weiskamp.