MANAJEMEN ARSITEKTUR GUDANG DATA UNTUK MENINGKATKAN PERFORMANSI SISTEM INFORMASI

DATA WAREHOUSE ARCHITECTURE MANAGEMENT TO IMPROVE INFORMATION SYSTEM PERFORMANCE

SURIANSYAH B

D082202025

PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2023

PENGAJUAN TESIS

MANAJEMEN ARSITEKTUR GUDANG DATA UNTUK MENINGKATKAN PERFORMANSI SISTEM INFORMASI

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi Magister Teknik Informatika

Disusun dan diajukan oleh

SURIANSYAH B

D082202025

Kepada

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2023

TESIS

MANAJEMEN ARSITEKTUR GUDANG DATA UNTUK MENINGKATKAN PERFORMANSI SISTEM INFORMASI

SURIANSYAH B
D082202025

Telah dipertahankan dihadapan Panitia Ujian Tesis yang dibentuk dalam rangka penyelesaian studi pada Program Magister Teknik Informatika Fakultas Teknik Universitas Hasanuddin
Pada tanggal 17 Mei 2023
Dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama

Dr.Ir.Amil Ahmad Ilham,S.T.,M.IT
NIP.197310101998021001

Pendamping Pembimbing

Dr.Eng.Ady Wahyudi Paundu, S.T.,M.T
NIP 19750313 200912 1 003

Dekan Fakultas Teknik
Universitas Hasanuddin

Prof. Dr. Eng. Ir. Muhammad Isran Ramli, ST., MT
NIP. 19730926 200012 1 002

Ketua Program Studi
S2 Teknik Informatika

Dr.Ir. ZahirZainuddin, M.Sc.
NIP. 19640427 198910 1 002

PERNYATAAN KEASLIAN TESIS

DAN PELIMPAHAN HAK CIPTA

Yang bertanda tangan dibawah ini :

Nama : Suriansyah B

Nomor Mahasiswa : D082202025

Program Studi : Teknik Informatika

Dengan ini menyatakan bahwa, tesis yang berjudul " Manajemen Arsitektur Gudang Data untuk Meningkatkan Performansi Sistem Informasi" adalah karya saya dengan arahan dari komisi pembimbing (Dr. Ir Amil Ahmad Ilham, S.T., M.IT dan Dr.Eng.Ady Wahyudi Paundu, S.T., M.T). Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apapun kepada perguruan tinggi manapun. Sumber informasi yang berasa atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam daftar pustaka tesis ini. Sebagian dari tesis ini telah dipublikasikan di Jurnal/Prosing (International Conference on Computer Science, Information Technology and Engineering (ICCoSITE 2023) Feb-2023) sebagai artikel dengan judul "*Optimization of Data Warehouse Architecture to Improve Information System Performance".*

Dengan ini saya limpahkan hak cipta dari karya tulis saya berupa tesis ini kepada Universitas Hasanuddin.

Gowa, 17 Mei 2023

Yang Menyatakan

Suriansyah B

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT. karena berkat rahmat dan karunia-Nya sehingga tesis yang berjudul "**MANAJEMEN ARSITEKTUR GUDANG DATA UNTUK MENINGKATKAN PERFORMANSI SISTEM INFORMASI**" ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-2 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan tesis ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tesis. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Tuhan Yang Maha Esa atas semua berkat, karunia, serta pertolongan-Nya yang tiada batas, yang diberikan kepada penulis disetiap langkah dalam pembuatan program hingga penulisan laporan tesis ini;
2. Kedua Orang tua penulis, Bapak Bahir dan Ibu St Aminah, S.Pd yang selalu menjadi motivasi terbesar dalam penyelesaian perkuliahan ini yang tidak pernah putus memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil;
3. Istri tercinta, Nurdinah,S.Pd yang dengan sangat sabar menemani dan memberikan semangat kepada penulis selama perkuliahan sampai penyusunan tesis, serta Bapak mertua saya Hatta,S.Pd dan Ibu Hilmawati dan anak sholeha saya Afifah Abidah S yang selalu mebakar semangat penulis;
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT selaku pembimbing I dan Bapak Dr. Eng. Ady Wahyudi Paundu, ST., M.T. selaku pembimbing II yang telah memberikan waktu, tenaga, pikiran, dukungan moril serta perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tesis;
5. Bapak Dr. Ir. Zahir Zainuddin, M.Sc ,Bapak Dr.Eng. Zulkifli Tahir, S.T., M.Sc. dan Ibu Dr. Ir. Ingrid Nurtanio, MT, Dr. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun sehingga laporan tesis ini menjadi lebih baik;
6. Para sahabat, teman-teman di Laboratorium *Computer Based System* serta teman seperjuangan di Laboratorium Animasi dan Mutimedia UNHAS yang telah memberikan begitu banyak bantuan, keceriaan dan pengalaman manis selama proses perkuliahan;

7. Teman-teman Pascasarjana UNHAS Angkatan 1-3 atas dukungan dan semangat yang diberikan selama ini;
8. Ibu Yuanita serta segenap Staf Departemen Magister Teknik Informatika yang telah banyak membantu penulis selama pengurusan administrasi;
9. Orang-orang terkasih yang tidak sempat dituliskan oleh penulis;

Akhir kata, penulis berharap semoga Allah SWT. Senantiasa berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tesis ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin ya Rabbal Alamin.

Wassalam

Gowa, 17 Mei 2023

Penulis

ABSTRAK

SURIANSYAH B. Manajemen Arsitektur Gudang Data untuk Meningkatkan Performansi Sistem Informasi. **(**dibimbing oleh **Amil Ahmad Ilham,** dan **Ady Wahyudi Paundu).**

Pertumbuhan data dari hari ke hari semakin meningkat, sehingga data yang tersimpan di data *warehouse* semakin menumpuk. Hal tersebut menyebabkan kecepatan *load* data pada sistem informasi menjadi berkurang, sehingga diperlukan manajemen pada data *warehouse* agar proses *load* menjadi lebih ringan meskipun pertumbuhan data semakin meningkat. Pada penelitian ini akan dibuat suatu algoritma penjadwalan pada *Hadoop* yang bertugas untuk mengeksekusi proses ekstraksi transformasi dan memuat data ringkasan ke dalam beberapa tabel, langka ini bertujuan untuk merampingkan dan mengoptimalkan proses *Extract*, *Transform*, *Load* (*ETL*) ke data *warehouse* dan mengurangi volume data dalam satu tabel kemudian data akan diindeks sesuai kunci utama di setiap tabel. Setelah dilakukan pengujian, dengan melakukan testing *query* pada arsitektur data *warehouse* yang di lakukan manajemen dan tidak di lakukan manajemen dengan baik, pada hasil yang di manajemen waktu *query* sebesar 1.418 detik, sedangkan tabel yang tidak dimanajemen dengan baik memiliki waktu query sebesar 2.418 detik. Serta pengujian kecepatan *load* data ke dalam sistem informasi dengan membandingkan *throughput* sistem yang dioptimalkan dan yang tidak dioptimalkan memiliki rata-rata perbedaan *throughput* sebesar 85%. Dengan hasil tersebut dapat disimpulkan bahwa penelitian ini berhasil meningkatkan kecepatan *load* data ke sistem informasi.

Kata Kunci: Gudang Data, Sistem Informasi, *ETL*, Manajemen, Penjadwalan

ABSTRACT

SURIANSYAH B. *Management of Data Warehouse Architecture to Improve Information System Performance*. **(***Supervised by* **Amil Ahmad Ilham,** *and* **Ady Wahyudi Paundu).**

The Data growth increasing daily, the amount of data stored in the data warehouse is growing rapidly. Slow performance is observed when loading queries from the data warehouse tables to the information system, as displaying data on the dasboard or information system requires accessing all the stored data. The speed of loading data on information better systems decreases, so optimization is needed in the data warehouse to make the load process lighter even though data growth is increasing. In this research, a scheduling algorithm will be created in Hadoop, which executes the transform extraction process and loads summary data into several tables. It aims to streamline and optimize the Extract, Transform, Load (ETL) process to the data warehouse and reduce the volume of data in one table, then will be indexed according to the primary key in each table so that when data is joined to several tables, it can be executed faster. After performing tests by querying data from different tables with the same goal, it was found that optimized tables result query time of 1.418 seconds.In contrast, unoptimized tables took longer with a query time of 2.418 seconds to achieve the same goal . For the testing of loading speed data into the information system by comparing the throughput of systems that are optimized and those that are unoptimized have an average throughput difference of 85%. With these results, it can be concluded that the speed in loading data into the information system has been successfully optimized by looking the comparison.

Keywords: *Data Warehouse, Information System, ETL,Management,Scheduling*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PERMASALAHAN DAN TUJUAN PENELITIAN

1.1 Latar Belakang

Di era digital seperti saat ini banyak perusahaan yang memanfaatkan teknologi untuk meningkatkan kinerja bisnisnya bahkan banyak perusahaan dalam menghadapi persaingan bisnisnya menggunakan sistem informasi atau *dasboard* untuk melakukan pemantuan performansi penjualannya dalam memperoleh keuntungan lebih besar, Hal ini mempengaruhi strategi penjualan yang akan di lakukan sebuah perusahaan. hadirnya sebuah *dasboard* sangat membantu dalam mengambil dan menyusun strategi penjualan kedepanya karena dengan memanfaatkan data *historis* yang di lakukan perusahaan akan menjadi lebih efisein dan optimal[1]

Namun semakin banyak transaksi dari sebuah perusahaan makin banyak data yang tersimpan sehingga data yang banyak itu, makin hari makin membebani proses *load* data ke *dasboard* ,sehingga performansi *dasboard* dalam hal ini menampilkan data kian hari makin melambat. Karena proses *query* data tiap kali sistem informasi di akses akan mengkalkulasi semua data yang tersimpan di tabel dimana data di simpan . hal itulah yang menyebabkan performance *dasboard* menjadi menurun.

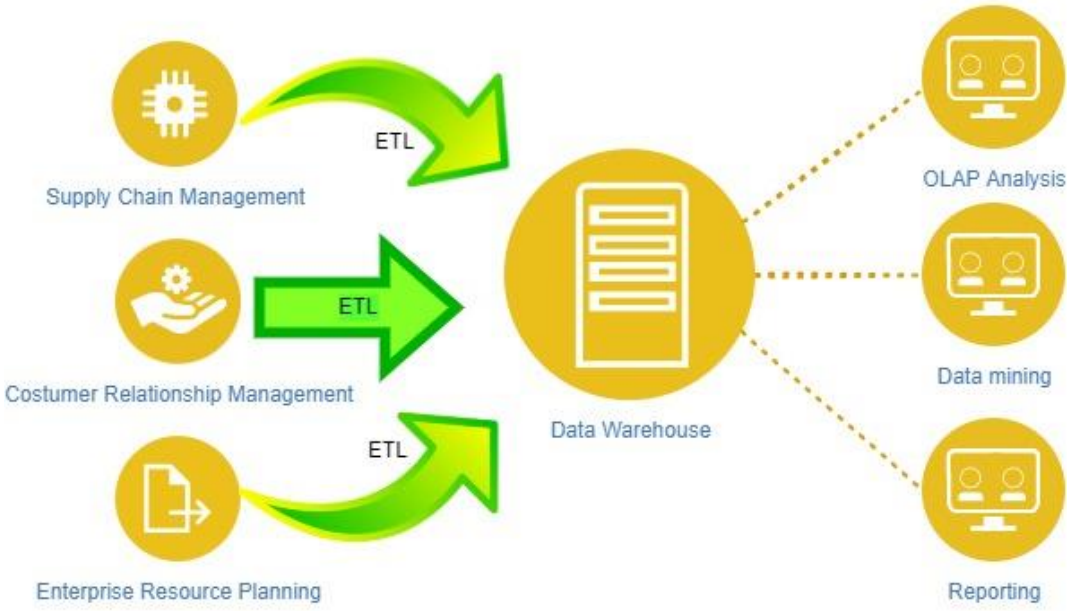
Dalam gudang data tardiri dari beberapa data base dari data *base* terdiri dari beberapa tabel dan dari tabel itu berisi data dari semua trasaksi yang dilakukan pelanggan . dari tabel itulah sistem informasi akan menampilkan dan

memvisualisasikan data. Sehingga penting untuk mamanejemen gudang data agar sistem informasi menjadi lebih cepat[2,3].

Namun saat data yang banyak di *load* ke dalam gudang data tentu akan memakan waktu dan *effort* yang lebih. dan saat ini proses *load* data ke gudang data dalam proses input data ke gudang data akan melaluai beberapa tahapan seperti (*ETL*) *Extract*, *Transform*, *Load* dari beberapa *source*. Dan saat data akan diload ke gudang data pihak data *egineering* akan malakukan *Query* sesuai kebutuhan analisis dalam hal ini masi konvensional sehingga proses update data di sistem informasi sangat tergantung dengan hasil *load* dari *query* hal itulah yang mempengarugi kecepatan update data di sistem informasi [4].

Ada tiga bentuk umum aristektur gudang data pertama *basic* dalam arsitektur gudang data *basic* ini biasanya di gunakan dalam perusahaan atau instansi dalam sekala kecil yang di mana data di olah masi tergolong sedikit. Yang kedua yaitu Arsitektur *With Staging Area* yag di mana Dan penerapan arsitektur gudang data ini di terapkan dalam perusahaan atau instansi yang data di olah sudah banyak sehingga data di kumpul terlebih dahulu dalam suatu *stagging* untuk di *cleansing* karena datanya banyak dan pastinya ada beberapa yang duplikat atau *null* akan debersihkan setelah itu data di tarik sesuai kebutuhan dan data di masukkan server yang di sebut gudang data .Dan yang ketiga *With Staging Area and Data Marts*.dalam hal ini peneliti akan mengmanajemen bentuk aristektur *with Staging area and data mart* yang dimana bentuk gudang data ini di aplikasikan untuk data yang tergolong besar dan terbagi beberapa data *base* dan memiliki data *center*

berikur gambaran arsitektur datanya.[5]:



Gambar 1. 1 Data Warehouse Arsitektur

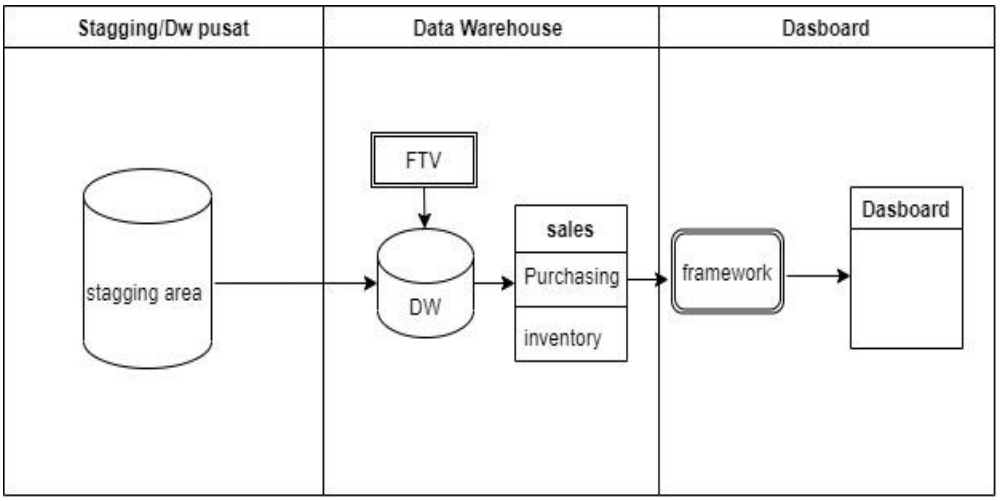
Terdapat masalah yang muncul ketika data di laod dalam jumlah besar yaitu membutuhkan *time cost* yang sangat besar seperti yang di sebutkan pada paper jurnal yang di tulis oleh Revathy Srekumar pada penelitiannya yang berjudul "*ETL Scheduling in Real-Time Data Warehousing*" [6] penelitian ini membahas mengenai aristektur gudang data ,*time cost* dan betapa pentingnya *scheduling*.maka diperlukanlah *scheduling* untuk proses *load* yang lebih efisien sebagai solusi dari permasalahan tersebut .

Penelitian ini juga merujuk pada penelitian yang di tulis oleh Agung Yudha Berliantara dan kawan kawannya yang berjudul "*Optimization Scheduling for Proses Extract, Transform, Load (ETL)* kelebihan dari penelitian meningkatkan kecepatan *load* data , namun bila data yang di load tidak di manajemen dengan baik

dan dioptimal maka data akan tetap menumpuk di tabel dan akhirnya proses *load* data ke sistem informasi akan tetap membebani dan akhirnya performansi sistem informasi menjadi menurun[7].

Dan penilitian ini juga merujuk pada penelitian yang di lakukan oleh Aloysius Adhyatma Herfangsyah dan kawan-kawan yang berjudul "Analisis Faktor Optimasi untuk Gudang data dengan Data Tabungan pada Bank XYZ" kelebihan dari manajemen yang dilakukan penggunaan partisi, *bucketing*, dan kompresi sehingga *time cost* yang di pakai saat proses *query* lebih cepat namun teknik ini saat di aplikasikan ke *framework* untuk menarik data ke sistem informasi kurang optimal karena saat data berjumlah jutaan *raw* akan tetap memakan *time cost* lebih lama[8].

Dari hasil observasi terhadap arsitektur data di telkomsel regional sulawesi sebagai tempat penelitian di perlihatkan pada gambar 2



Gambar 1. 2 Arsitektur alur data

Dari alur data pada gambar 1.2 terlihat bahwa data dari *stagging area* ditarik berdasarkan kebutuhan data *reporting* untuk sistem informasi yang di mana proses *query* data di lakukan setiap data di *dasboard* akan di *update* , dan data di masukkan

ke gudang data lokal dan diolah oleh *framework* untuk di *load* ke sistem informasi. Yang dimana saat data tersimpan di tabel gudang data banyak proses *load* data ke sistem informasi menjadi lambat [9]

1.2 Perumusan Masalah

Berdasarkan latar belakang maka rumusan masalah pada penelitian ini adalah :

1. Bagaimana memanajement arsitektur gudang data agar proses *load* data dari gudang data ke sistem informasi meningkat.?

2. Bagaimana mengefesiensikan serta mengautomatisasi proses *load* data ke dalam data *warehouse* ?

1.3 Tujuan Penelitian

Adapun tujuan yang akan dicapai pada penelitian ini:

1. Meningkatkan *performance* sistem informasi
2. Mengefisiensi serta mengautomatisasi proses *load* data ke data *warehouse*

1.4 Manfaat Penelitian

Manfaat dari penelitian adalah:

1. Meningkatkan proses *load* data ke sistem infromasi sehingga sistem informasi ini lebih cepat saat di akses

2. Mengautomatisasi update data secara scheduling ke gudang data.

3. Meningkatkan proses *query* data dari tabel gudang data.

1.5 Batasan Masalah

Adapun batasan masalah penelitian ini :

1. Dalam penelitian ini, fokus penelitian terbatas pada peningkatan *performance* sistem informasi
2. Manajemen arsitektur gudang data.

BAB II

KAJIAN LITERATUR DAN METODE PENYELESAIAN MASALAH

2.1 Kajian Pustaka

Beberapa penelitian terkait yaitu, Agung Yudha Berliantara & Satrio Agung Wicaksono,Aryo Pinnadito pada tahun 2017, Manajemen *Scheduling* untuk *Proses Extract, Transform, Load (ETL)* pada Gudang data Menggunakan Metode Round Robin Data Partitioning.Penelitian ini menjelaskan penerapan *scheduling* pada gudang data menggunakan *methode round robin data partitioning*, yang di mana perancangan skema multidimensi gudang data di sesuaikan dengan kebutuhan yang di mana method round robin ini data akan di bagi kedalam perisi partisi yang telah di sediakan sehingga waktu yang di gunakan untuk memproses data yang masuk ke tabel target akan lebih sedikit. Namun dari penelitian ini saat data banyak di laod ke tabel target yang di mana transaksi setiap hari kian ahari akan memakan storage gudang data dan akhirnya mebebani proses *load* saat data di *load* ke *dasboard*[9,10,11].

Aloysius Adhyatma Herfangsyah, Willy Sudiarto Raharjo, Antonius Rachmat Chrismanto. pada tahun 2020 “Analisis Faktor Manajemen untuk Gudang data dengan Data Tabungan pada Bank XYZ”, Manajemen yang dilakukan dengan cara analisis penggunaan partisi, *bucketing*, dan kompresi , Perlakuan partisi dan bucketing pada tabel akan memberikan waktu query yang lebih cepat. Kompresi data akan mengecilkan ukuran data pada ruang penyimpanan tetapi ada kelemahan dari kompresi ini, saat data di kompresi, rata-rata waktu *query* yang dibutuhkan akan menjadi lebih lama. Basis data yang digunakan pada penelitian ini berfokus

pada *Hive*, yaitu basis data yang ditujukan untuk melakukan proses analisis data, terdapat contoh lain basis data pada ekosistem *Hadoop*,seperti penggunaan *HBase*, *HBase* berfokus pada *real time querying*, yang ideal digunakan jika memerlukan data yang acak saat *write* atau *read*, sehingga perlu dilakukan penelitian faktor manajemen lebih lanjut tentang basis data *HBase* atau basis data lain yang terintegrasi dengan *Hadoop*, sehingga pada penelitian lain, dapat dipilih basis data yang sesuai dan dibuat secara optimal untuk keperluan pemrosesan data secara efisien dan cepat[11,12.13]

I Nyoman Adnyana Putra, Rukmi Sari Hartati, Ni Wayan Sri Aryani, pada tahun 2019 “Manajemen Proses ETL dengan Methode Heuristik untuk membangun Gudang data”. Manajemen dalam suatu proses extract yang menerapakan beberapa teknik guna menangani permasalahan pada proses ETL yang tujuannya mengekstraksi seluruh data yang menyebabkan banyak data yang harus diproses. Teknik yang digunakan pada proses ini adalah mendeteksi data yang berubah sehingga tidak mengekstraksi seluruh data sehingga jumlah data yang seharusnya diproses menjadi semakin efisien. Untuk dapat mempercepat waktu saat proses ETL, digunakan suatu indeks pada database sumber sehingga proses seleksi data menjadi semakin cepat dan proses. Manajemen pada proses transform dilakukan penerapan suatu teknik denormalisasi untuk membuat tabel yang ada diterapkan pula pengoptimalan penulisan aljabar relasional dalam proses denormalisasi dan dalam pembuatan tabel dimensi agar waktu proses menjadi semakin cepat. Kesimpulan dari penelitian di atas fokus pada peroses ETL ke gudang data yang di mana data tetap di *load* ke gudang data sehingga pada batas tertentu data akan

semakin banyak yang akhirnya mebebani proses load saat data akan di tampilkan ke sistem informasi [14,15,16,17,18].

2.2 Hadoop Distributed File System (HDFS)

HDFS adalah open source project yang dikembangkan oleh *Apache Software Foundation* dan merupakan subproject dari *Apache Hadoop*. *Apache* mengembangkan HDFS berdasarkan konsep dari *Google File System (GFS)* dan oleh karenanya sangat mirip dengan GFS baik ditinjau dari konsep logika, struktur fisik, maupun cara kerjanya. Sebagai layer penyimpanan data di *Hadoop*, HDFS adalah sebuah sistem file berbasis Java yang *fault-tolerant*, terdistribusi, dan scalable. Dirancang agar dapat diaplikasikan pada kluster dan dapat dijalankan dengan menggunakan proprietary atau *commodity server*. HDFS ini pada dasarnya adalah sebuah direktori dimana data disimpan yang bekerja sesuai dengan spesifikasi dari *Hadoop*. Data tersimpan dalam kluster yang terdiri dari banyak node komputer/server yang masing-masing sudah terinstalasi *Hadoop*.

Sistem penyimpanan terdistribusi pada HDFS melakukan proses pemecahan file besar menjadi bagian-bagian lebih kecil dan kemudian didistribusikan ke kluster-kluster sehingga memungkinkan pemrosesan secara pararel. HDFS memiliki banyak kesamaan dengan sistem file terdistribusi lainnya, nanum perbedaan yang terutama adalah model *Write-Once-Read-Many (WORM)* pada HDFS yang melonggarkan persyaratan kontrol konkurensi, menyederhanakan koherensi data, dan memungkinkan akses *throughput* yang tinggi. *HDFS* memiliki fitur-fitur sebagai berikut:

a. Sangat sesuai untuk penyimpanan, pengelolaan dan pemrosesan dataset yang besar secara terdistribusi.

b. *Hadoop* menyediakan antarmuka perintah untuk berinteraksi dengan HDFS.

c. *Heartbeat* memudahkan pemeriksaan status kluster.

d. Akses data melalui *MapReduce streaming*.

e. HDFS menyediakan *file permissions and authentication*.

f. *Fault detection dan recovery*.

g.Lokasi komputasi berada dekat dengan data untuk mengurangi *traffic* jaringan dan meningkatkan *throughput*.

h. Model data dan struktur HDFS

Sebagai *distributed file system, HDFS* menyimpan suatu data dengan cara membaginya menjadi potong-potongan data yang disebut blok berukuran 64 MB dan kemudian disimpan pada node-node yang tersebar dalam kluster. Ukuran blok tidak terpaku pada nilai tertentu sehingga dapat diatur sesuai kebutuhan. Walaupun data disimpan secara tersebar, namun dari sudut pandang pengguna, data tetap terlihat utuh dan diperlakukan seperti halnya mengakses *file* pada satu media penyimpanan. Berbeda dengan sistem *file* pada umumnya, *HDFS* dapat bertumbuh tanpa batas, karena secara arsitektur dan administrasinya dapat menambah jumlah *node* sesuai kebutuhan. Abstraksi satu file yang berada di beberapa *node* memungkinkan ukuran *file* bertumbuh tanpa batas.

HDFS memiliki komponen utama yaitu *namenode* dan *datanode*. Namenode adalah sebuah node yang bertindak sebagai master, sedangkan datanode adalah node-node dalam kluster yang bertindak sebagai *slave*. *Namenode* bertanggung-jawab menyimpan, mengorganisir dan *mengontrol blok-blok* data yang disimpan pada *node-node* yang tersebar dalam kluster. Datanode bertanggung-jawab menyimpan *blok-blok* data yang ditujukan kepadanya, dan

secara berkala melaporkan kondisinya kepada *namenode*. Jadi, *namenode* seperti manager yang mengatur dan mengendalikan kluster. Sedangkan, *datanode* seperti *worker* yang bertugas menyimpan data dan melaksanakan perintah dari *namenode*.

Setiap data yang disimpan pada *HDFS* memiliki lebih dari satu salinan, yang disebut sebagai *Replication Factor (RF)*. Secara default nilai RF adalah 3, yang berarti satu file tersimpan di 3 *datanode* berbeda sehingga jika salah satu datanode rusak, maka file dapat diperoleh dari datanode lain. *Datanode* mengirimkan sinyal setiap 3 detik yang disebut heartbeat kepada namenode untuk menunjukkan bahwa *datanode* tersebut masih aktif. Apabila dalam 10 menit namenode tidak menerima heartbeat dari datanode, maka datanode tersebut dianggap rusak atau tidak berfungsi sehingga setiap permintaan baca/tulis dialihkan ke *node* lain. Dengan heartbeat, maka namenode dapat mengetahui dan menguasai kondisi kluster secara keseluruhan. Sebagai respon atas heartbeat dari datanode, selanjutnya namenode akan mengirimkan perintah kepada datanode[19,20,21].

2.3 Apache hive

Apache hive adalah Perangkat lunak data *warehouse* yang memfasilitasi dalam membaca, menulis, dan mengelola set data besar yang berada di penyimpanan terdistribusi menggunakan *SQL*. Struktur tersebut dapat diproyeksikan ke data yang sudah ada di penyimpanan. Dengan kata lain, *Hive* adalah sistem open sources yang memproses data terstruktur di *Hadoop*, berada di lapisan paling atas dan terakhir untuk meringkas *Big Data*, serta memfasilitasi analisis serta *queri*[22,23].

Berikut ini adalah karakteristik utama *Hive* yang perlu diingat saat menggunakannya untuk pemrosesan data:

1. *Hive* dirancang untuk melakukan *query* dan mengelola hanya untuk data terstruktur yang disimpan dalam tabel
2. *Hive* dapat diskalakan, cepat, dan menggunakan konsep yang telah familiar
3. Skema akan disimpan dalam database, sementara data yang diproses masuk ke *Hadoop Distributed File System (HDFS)*
4. Tabel dan database dibuat terlebih dahulu; lalu data akan dimuat ke dalam tabel yang sesuai
5. *Hive* mendukung empat format file: *ORC, SEQUENCEFILE, RCFILE (Record Columnar File)*, dan *TEXTFILE*
6. *Hive* menggunakan bahasa yang terinspirasi dari *SQL*, sehingga memudahkan pengguna dalam melakukan pekerjaan yang berhubungan dengan kompleksitas pemrograman *MapReduce*. Ini membuat pembelajaran lebih mudah diakses dengan menggunakan konsep yang familiar dan ditemukan dalam database relasional, seperti kolom, tabel, baris, dan skema, dan lain-lain.
7. Perbedaan paling signifikan antara *Hive Query Language (HQL)* dan *SQL* adalah bahwa *Hive* menjalankan kueri pada infrastruktur *Hadoop* disamping infrastruktur database tradisional
8. Karena pemrograman *Hadoop* bekerja pada file yang datar, maka *Hive* menggunakan struktur direktori untuk data 'partisi', sehingga meningkatkan kinerja pada kueri tertentu
9. *Hive* mendukung *partisi* dan *bucket* untuk pengambilan data yang cepat dan sederhana

10. *Hive* mendukung *custom user-defined functions* (UDF) untuk beberapa tugas seperti pembersihan dan pemfilteran data. *HIVE UDF* dapat didefinisikan sesuai dengan persyaratan programmer.

Dengan begitu algoritma yang kami akan aplikasikan berada dalam apache hive ini yang di mana berisi query serta konfigurasi dalam menarik serta meload data dari HDFS.[24,25] .

2.4 State of the Art Penelitian

Penelitian-penelitian terkait yang telah dilakukan sebelumnya dapat dilihat pada Tabel **2.1** dibawah ini.

Tabel **2.1** State of the Art

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
1.	**Judul**: Management Arsitekture gudang data untuk Mengoptimalkan Performance Dasboard (Topik yang diusulkan) **Penulis :** -	**Objek :** load data ke gudang data dan manegemet arsitekture gudang data **Permasalahan :** Bagaimana meload data ke gudang data secara optimal serta memanagement aritekture gudang data agar performance dasboard menjadi meningkat	Membuat algoritma pada Shell Script (sh) yang di tampung oleh cron agar bisa di gunakan dalam penarikan data dari warehouse pusat dan di load ke gudang data local, serta membagi beberapa tabel dalam aristekture gudang data agar dapat menampung data summary	Kinerja topik yang diusulkan akan mampu mengoptimalkan proses load data saat dasboard di akses
2.	**Judul** : Analisis Faktor Manajemen untuk Gudang data dengan Data Tabungan pada Bank XYZ	**Objeck** : Gudang data **Permasalah** : Pemprosesan data di warehouse kurang optimal dikarenakan	Penggunaan partisi, bucketing, dan kompresi, untuk mengoptimalkan kinerja query	

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
	Penulis : Aloysius Adhyatma Herfangsyah, Willy Sudiarto Raharjo, Antonius Rachmat Chrismanto Tahun : 2020 **Penerbit :**JUTEI	bertambahnya komlesitas data yang terus bertambah		Penyelesaian proses query menjadi meningkat
3.	Judul : Manajemen Proses ETL dengan Methode Heuristik untuk membangun Data Warehouse Penulis : I Nyoman Adnyana Putra, Rukmi Sari Hartati, Ni Wayan Sri Aryani Tahun : 2019 Penerbit : JST (Jurnal Sains dan Teknologi)	**Objeck** : Gudang data Extrak,Transfor,Load (ETL) **Permasalah** : pengumpulan data yang teritegrasi dari beberapa sumber terus bertambah sehingga kinerja dari proses ETL akan semakin lama dan semakin berat	Menerapkan Methode Heuristik yang di mana saat Prsoses ETL dilakukan akan mengubah mekanisme alur kerja ETL	Proses ETL ke gudang data menjadi lebih cepat dan efisien
4.	Judul : Incremental updates using Data Warehouse versus Data Marts[5] Penulis : Sonali Chakraborty, Jyotika Doshi Tahun : 2018 Penerbit : IEEE	**Objeck:** Gudang data, Data Marts **Permasalah:** Waktu pengambilan hasil Query dari gudang data memakan waktu lama karena ukuran data cukup besar	Menggunakan pendekatan basis data MQDB (Materialozed Query Database)	Mengurangi waktu pemprosesan query saat query lakukan pada data mart

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
5.	Judul : Resource Efficient data warehouse Optimization[6] Penulis : Illia Sokolov, Ihor Turkin Tahun : 2018 Penerbit : IEEE	**Objeck:** Gudang data **Permasalah:** Waktu yang di butuhkan dalam mengolah data yang bervolume besar sangat lama	Melakukan pengideksan untuk menghindari volume data yang besar serta Melakukan menyortiran melakukan menyimpanan data dalam urutan yang di perlukan untuk mempercepat query	Meningkatkan proses query pada gudang data
6.	Judul **:** Compact Data Structures to Represent and Query data warehouse into Main Memory[7] Penulis : C. Vallejos, M. Caniupán, and G. Gutiérrez Tahun : 2018 Penerbit : IEEE	**Objeck:** Gudang data,memory **Permasalahan:** Saat proses query berlangsung dan jumlah data yang di query banyak akan memakanan time cost lebih lama untuk dapat melihat hasil dari query	Dengan mengunakan struktur data yang sama akan merepresentasikan gudang data dalam peningkatan ruang di memory utama sehingga proses query dapat lebih cepat	Meningkatkan proses query pada gudang data
7.	Judul : Integrated Architecture of Data Warehouse with Business Intelligence Technologies[8] Penulis: Ch Anwar ul Hassan, Rizwana Irfan, Munam Ali Shah Tahun: 2018 Penerbit:IEEE	**Objeck:** Gudang data,Bisnis Intelligence **Permasalahan:** Proses pengambilan keputusan tergolong lama karena arsitekture gudang data yang traditional sehingga belum bisa menganalisa data dari gudang data itu sendiri	Dalam gudang data akan melakukan proses pemodelan data untuk menganalisa setiap keterkaitan data	Pengambilan keputusan di harapkan dapat lebih cepat

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
8.	Judul : Implementation of Database Massively Parallel Processing System to Build Scalability on Procces Data Warehouse[9] Penulis: Fajar Ciputra Daeng Bani, Suharjito, Abba Suganda Girsang, Diana Tahun: 2018 Penerbit:ElSEVIER	**Objeck:** Gudang data **Permasalahan:** Data transaksi lebih banyak di banding tabel sumber yang ada hal ini membuat pelaporan bisnis kurang efisien dan membanjiri proses query di gudang data sehingga tidak memenuhi persyratan bisnis	Mengiplementasikan Massive Paralel Processing (MPP) pada Gudang data dan data base Greenplum dengan cara mengukur komplesitas query di gudang data	Pengguna atau stakeholder dapat lebih cepat dan optimal dalam mendapatkan laporan
9.	Judul : Efficient Data Acces And Performance Improvement Model For Virtual Data Warehouse[10] Penulis: Fakhri Alam Khan,Awais Ahamad, Muhammad Imran dkk Tahun: 2017 Penerbit:ElSEVIER	**Objeck:** Model Virtual Gudang data **Permasalahan:** Pengumpulan data bervolume besar sangat menghambat kinerja operasional untuk mendapatkan laporan dari perusahaan serta mahalnya alat bila ingin membangun sebuah Gudang data	Mengusulkan konsep yang di sebut Efficient Data Acces dalam Virtual Gudang data	Mendapatkan laporan dari proses bisnis yang berjalan dengan menggunkan Virtual Gudang data
10.	Judul : Manajemen Scheduling untuk Proses Extract, Transform, Load (ETL) pada Gudang data Menggunakan Metode Round Robin Data Partitioning	**Objeck:** Gudang data **Permasalahan:** Proses Scheduling Saat dilakukan ETL membutuhkan time-cost yang besar sehingga ketidakkonsistenya data yang siap di Gudang data	Mengusulkan methode Roud-Robin untuk mengifisiensikan Proses ETL ke Gudang data yang di mana tabel tujuan akan di bagi menjadi beberapa bagian	Dengan diterapkannya Methode Round-Robin ini memberikan waktu eksekusi lebih efisien hingga 60,1 berdasarkan jumlah data yang di olah

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
	Penulis: Agung Yudha Berliantara,Satrio Agung Wicaksono,Aryo Pinnadito Tahun: 2017 Penerbit:JPTIIK			
11	Judul : A Scheduling System for Big Data Hybrid Computing Workflow Penulis: Yongbo Zhu, Haihong E and Meina Song Tahun: 2016 Penerbit:IEEE	**Objeck:** *Scheduling Big data* **Permasalahan:** Proses Scheduling Saat dilakukan *ETL* membutuhkan time-*cost* yang lama	Menggabungkan proses *ETL Workflow* saat penjadwalan *ETL* bekerja	Dengan menggabungkan proses c*omputing* kinerja saat *ELT* berlangsung berhasil meningkatkan kecepatan
12	Judul **:** ETL-aware Materialized View Selection in Semantic Data Stream Warehouses Penulis : Nabila Berkani, Ladje Bellatreche, Carlos Ordonez Tahun : 2018 Penerbit : IEEE	**Objeck:** *Scheduling Big data* **Permasalahan:** Proses pengambilan keputusan tergolong lama karena arsitekture gudang data yang traditional sehingga belum bisa menganalisa data	Proses ETL menggunkana semantic unutk melihat data yang ada pada data warehouse	Metode ini dapata meringankan saat data akan di akses
13	Judul**:** Improved Heuristic Job Scheduling Method to Enhance	**Objeck:** *Scheduling , data analytic* **Permasalahan:**	Menginprovisasi *scheduling Heuristic* untuk meringan kan data yang akan di load	*Throughtput* data saat data besar di load dapat leih cepat meskipun

No	Judul Karya Ilmiah, Nama, Tahun Terbit Dan Penerbit	Objek Dan Permasalahan	Metode Penyelesaian	Kinerja
	Throughput for Big Data Analytics Penulis : Zhiyao Hu and Dongsheng Li Tahun : 2020 Penerbit : IEEE	Waktu yang di butuhkan dalam mengolah data yang bervolume besar sangat lama		volume data besar
14	Judul **:** A Holistic View of Data Warehousing in Education Penulis : Ján Cigánek Tahun : 2018 Penerbit : IEEE	**Objeck:** *Data warehouse*, *education* **Permasalahan:** Proses pengambilan keputusan tergolong lama karena gudang data yang traditional banyak menyimpan data sehingga membebani saat pencarian suatu data	Dalam *data warehouse* akan melakukan proses pemodelan data untuk menganalisa setiap keterkaitan data	Pengambilan dan pencarian data dapat meningkatkan pencarian dan akses data
15	Judul : Proposed Techniques to Design Speed Efficient Data Warehouse Architecture for Fastening Knowledge Discovery Process Penulis: Abhishek Gupta, Arun Sahayadhas, Vivek Gupta Tahun: 2020 Penerbit:IEEE	**Objeck:** Gudang data **Permasalahan:** Data yang tersimpan pada data warehouse sangat banyak, saat di lakukan analisis data di butuhkan waktu untuk mendapatkan hasil analisisnya	Melakukan partisi data serta pengideksan data di setiap tabel iduk agar saat di lakukan penarikan data untuk di analisis dapat lebih cepat untuk mengeluarkan hasil query	Dengan di lakukan partisi data dan index kinerja data warehouse lebih cepat

Berdasarkan uraian beberapa keaslian penelitian pada tabel di atas, maka pada penelitian ini akan Mengmanajemen proses load data ke *dasboard* dan memanajement arsitekture gudang data dengan lebih optimal untuk meningkatkan proses load data saat dasboard di akses . Hasil pengujian dari setiap skenario akan dianalisa untuk mengetahui sejauh mana performansi yang dihasilkan dari teknik yang diterapkan.

2.5 Target Hasil Penelitian

Berdasarkan penelitian yang akan dilakukan maka target penelitian yang diharapkan adalah membuat algoritma untuk menarik data dari gudang data pusat dan data di load ke gudang data local serta mamanagement arsitekture dari gudang data local selanjutanya membagi beberapa tabel untuk menampung data summary agar mengurangi beban pada query saat proses load data ke sistem informasi, serta mempercepat proses transmisi data ke sistem informasi yang digunakan, yang pada akhirnya performance dasboard menjadi lebih cepat.

2.6 Kerangka Pikir

Dibawah ini merupakan kerangka pemikiran dari penelitian tentang "Management Arsitekture Gudang Data untuk Mengoptimalkan Performansi Sistem Informasi"

Rumusan Masalah

- Bagaimana memanajemen arsitekture gudang data agar proses *load* data ke *dasboard* meningkat.?
- Bagaimana agar data terupdate setiap harinya ke dasboard.?

Tujuan Penelitian

- Meningkatkan *performance Dasboard*
- Mengmanajemen management dan Arsitektur gudang data.
- Mengautomatisasi load data ke Gudang data

Methode Penyelesaian

Membuat algoritma pada Shell Script (sh) yang di tampung oleh cron agar bisa di gunakan dalam penarikan data dari warehouse pusat dan di load ke gudang data local, serta membagi beberapa tabel dalam aristekture gudang data agar dapat menampung data summary

Hasil Yang Diharapkan

Dengan Menagement Arsitekture Gudang data secara Optimal di Harapkan akan Meningkatkan Performance Sistem informasi