Implementasi Hadoop Framework: Analisis Data Nasabah Bank Dengan Metode Mapreduce

Indah Gala Putri*, Zahra Hanifa, Marda Haryani, Fajar Pradika, Asmel Aritonang
Jurusan Sistem Komputer, Universitas Sriwijaya, Indonesia

*Korespondensi: 09011182126033@student.unsri.ac.id

ARTICLE INFO

Article History:
- Received 23 July 2023
- Received in revised form 22 August 2023
- Accepted 10 September 2023
- Available online 30 October 2023

ABSTRAK

Pesatnya perkembangan tekologi di era sekarang membuat pertumbuhan dan pengolahan data menjadi semakin besar, sehingga sangat diperlukan suatu metode dalam pengolahan data yang berukuran besar atau disebut dengan *big data. Big data* mempunyai karakteristik data yang variatif, pertumbuhan data yang cepat, dan kompleks. Maka dari itu pengimplementasian Hadoop Framework dalam mengelolah *big data* sangatlah dibutuhkan sebagai media untuk pengambilan keputusan, pemrosesan informasi, dan otomatisasi proses. Pada jurnal ini akan dibahas tentang implementasi Hadoop dalam pengolahan big data secara multinode yaitu dengan membagi akses data menjadi master dan slave. Di mana master akan menjalankan layanan NameNode dan slave akan menjalankan layanan DataNode. Node slave akan terhubung ke node master, dan mereka bekerja sama untuk memproses data dan menjalankan tugas-tugas pemrosesan secara terdistribusi. Ini memungkinkan pemrosesan data yang besar dan skalabel karena tugas-tugas dapat dibagi di antara banyak node slave. Dengan begitu data dapat diolah dengan efektif dan efisien. Adapun dalam melakukan analisis, dataset Nasabah Bank akan di kelompokkan dengan menggunakan metode MapReduce pada Hadoop Distributed File System (HDFS) untuk mengetahui jumlah transaksi yang terjadi setiap bulannya.

Kata Kunci: Big Data, Hadoop, HDFS, MapReduce, Nasabah Bank

ABSTRACT

The rapid development of technology in the current era has led to the significant growth and processing of data, necessitating a method for handling large-scale data, known as big data. Big data exhibits characteristics such as varied data, rapid data growth, and complexity. Therefore, the implementation of the Hadoop Framework in managing big data is crucial as a platform for decision-making, information processing, and process automation. This journal discusses the implementation of Hadoop in multi-node big data processing, dividing data access into master and slave nodes. The master executes the NameNode service, while the slave runs the DataNode service. Slave nodes connect to the master node, collaboratively processing data and executing distributed processing tasks. This enables large and scalable data processing as tasks can be distributed among many slave nodes, allowing for effective and efficient data processing. In the analysis, the dataset of Bank Customers will be grouped using the MapReduce method on the Hadoop Distributed File System (HDFS) to determine the number of transactions occurring each month.

Keywords: Big Data, Hadoop, HDFS, MapReduce, Bank Customers

1. PENDAHULUAN

Di era digital yang berkembang pesat saat ini, big data telah menjadi tren dalam dunia teknologi informasi saat ini. *Big data* merupakan sumber data dengan volume yang besar, banyak variasi dan memiliki kecepatan data yang cepat [1]. Menurut *Statistical Analysis System* (SAS), *big data* adalah istilah yang sering digunakan untuk menggambarkan perkembangan dan ketersediaan data terstruktur atau tidak terstruktur. *Big data* juga dapat digunakan untuk menjelaskan tentang tren yang terjadi saat ini [2].

Data telah menjadi salah satu aset paling berharga bagi perusahaan dan dunia usaha, termasuk sektor perbankan. Analisis data nasabah bank merupakan langkah penting yang membantu bank untuk mengambil keputusan yang akurat dan benar berdasarkan data, membantu meningkatkan citra perusahaan di mata nasabah, dan membuat rencana tindakan dengan mengetahui perilaku nasabah serta mengetahui tren pasar dan keinginan nasabah [3]. Hal ini memperkuat fakta bahwa pertumbuhan data memerlukan perencanaan profesional dan staf khusus [4]. Maka dari itu pengolahan terhadap *big data* merupakan suatu hal yang kritis. Pengolahan terhadap big data juga bukan merupakan suatu hal yang mudah. Pengolahan *big data* tidak dapat disamakan dengan pengolahan data dengan ukuran yang relatif kecil. Sehingga tidak memungkinkan untuk diproses menggunakan perangkat pengelola database konvensional ataupun aplikasi pemrosesan data lainnya [5]. Single Komputer akan mengalami kinerja yang buruk atau tidak dapat memproses data jika ukuran datanya melebihi kapasitas memori komputer. Oleh karena itu diperlukan suatu alat atau kerangka kerja yang dapat mendukung pengolahan *big data.*

Untuk mengatasi tantangan yang terkait dengan kompleksitas penerapan *big data*, *Apache Software Foundation* (ASF) menciptakan *Apache Hadoop* [6]. Hadoop merupakan suatu kerangka kerja yang digunakan untuk mengolah data berukuran besar dimana Hadoop mempunyai sistem khusus untuk menunjang kinerja komputer yang dirancang untuk menyimpan dan menganalisis data berukuran besar[7]. *Apache Hadoop* memiliki dua fitur utama yaitu *Hadoop MapReduce* yang merupakan *framework Mapreduce* dan *Hadoop Distributed File System* (HDFS) untuk sistem file terdistribusi [8]. *Hadoop MapReduce* menggunakan HDFS untuk mengakses segmen file dan menyimpan hasil yang digunkan oleh aplikasi *Hadoop* [9]. Adapun untuk arsitektur HDFS bersifat Master-Slave. Di mana *Hadoop* terdiri atas NameNode yang bertanggung jawab dalam mengelola sistem file, dan DataNode yang bertanggung jawab dalam mengelola penyimpanan data di setiap node [10].

2. TINJAUAN PUSTAKA

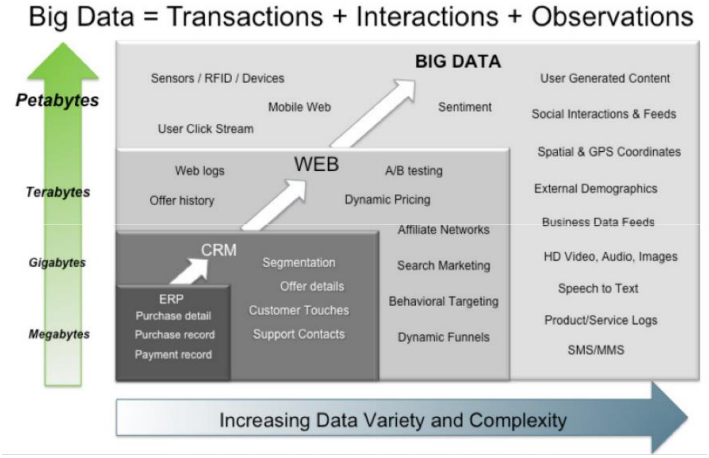
2.1 Big Data

Big Data adalah istilah yang menggambarkan ukuran data dalam jumlah yang besar, baik terstruktur maupun tidak terstruktur [11]. *Big Data* pertama kali dicetuskan oleh seorang analis industri bernama Doug Laney yang memaparkan 3 permasalahan *Big Data* atau yang sering disebut dengan “The 3Vs of Data” [12]. Adapun yang termasuk 3V itu diantaranya :

- *Volume*. Organisasi mengumpulkan data dari berbagai sumber, termasuk transaksi komersial, media sosial, dan informasi dari sensor atau mesin. Di masa lalu, jenis pekerjaan ini menimbulkan masalah, namun teknologi baru (seperti Hadoop) dapat mengurangi masalah ini [11].
- *Velocity*. Aliran data harus dikelola dengan cepat dan efisien melalui perangkat keras dan perangkat lunak. Teknologi perangkat keras seperti tag RFID dan sensor pintar lainnya juga diperlukan untuk menangani data waktu nyata [11].

- *Varietas*. Data yang dikumpulkan memiliki format dan jenis yang berbeda-beda. Mulai dari yang terstruktur, data numerik dalam database tradisional, data dokumen terstruktur teks, email, video, audio, transaksi keuangan dan banyak lagi [11].

Big Data merupakan kumpulan data dengan volume yang besar dan berasal dari berbagai jenis sumber data di seluruh dunia yang dapat diakses dimanapun dan kapan pun serta dapat bertambah dengan pesat [13]. Peningkatan volume, velositas dan variasi data banyak diakibatkan oleh adopsi internet. Setiap orang menciptakan konten atau setidaknya meninggalkan jejak digital yang dapat digunakan untuk inovasi. Isi dari *Big Data* terdiri dari transaksi, interaksi dan monitoring atau bisa dikatakan apa saja yang berhubungan dengan jaringan internet, jaringan komunikasi dan jaringan satelit seperti gambar di bawah ini [12].

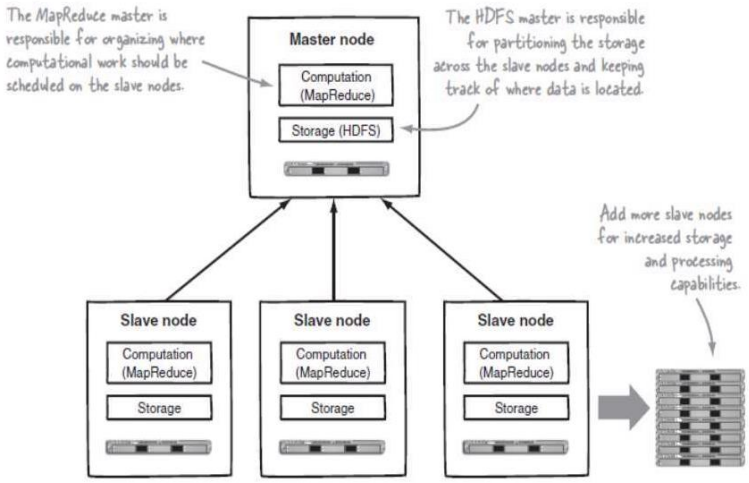


Gambar 1. Isi Big Data

Dari gambar diatas, kita dapat menyimpulkan peranan *Big Data* dalam berbagai urusan kehidupan kita cukup krusial. *Big Data* memainkan peran kunci dalam memahami dan mengelola data yang dihasilkan oleh adopsi internet yang pesat, dengan potensi untuk menghasilkan wawasan yang berharga dan menginformasikan berbagai aspek kehidupan dan bisnis.

2.2 Apache Hadoop

Apache Hadoop adalah sebuah perangkat kerangka kerja yang *open source* untuk memproses data dengan jumlah yang besar dengan memanfaatkan pemrosesan secara paralel [14]. Hadoop awalnya dirancang untuk memecahkan masalah skalabilitas yang ada pada Nutch (sebuah *open source* dan *search engine*) [2]. Hadoop bekerja di lingkungan yang menyediakan penyimpanan dan komputasi terdistribusi di sekelompok komputer [7]. Hadoop terdiri atas NameNode dan beberapa DataNode [15]. Hadoop juga merupakan *master-slave architecture*. Adapun berikut arsitektur dari Hadoop.



Gambar 2. Arsitektur Hadoop

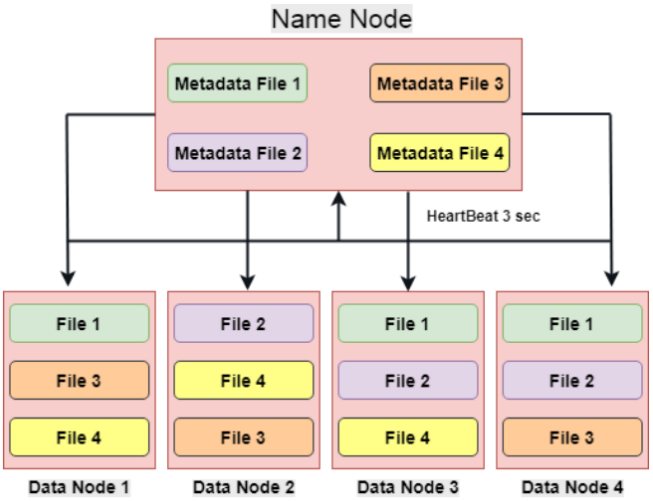
Pada Gambar 2. di atas terlihat bahwa Hadoop memiliki master node dan slave node yang terdiri dari *Hadoop Distributed File System* (HDFS) untuk media penyimpanan serta MapReduce untuk kemampuan komputasi. Fitur yang melekat pada Hadoop adalah segmentasi data dan komputasi paralel dari sumber data yang besar. Media penyimpanan tersebut dan skala kemampuan komputasi dengan penambahan hosts untuk Hadoop cluster, serta dapat mencapai ukuran volume hingga petabytes pada clusters dengan ribuan hosts [2]. Selain itu Apache Hadoop juga terdiri dari beberapa modul antara lain [14] :

- *Hadoop Common* yang memuat library dan file – file dasar untuk keperluan modul lainnya.
- *Hadoop Distributed File System* (HDFS) sebagai file system yang terdistribusi untuk men-support data dengan *bandwith* yang besar.
- Hadoop YARN sebagai resource management platform yang mengatur pembagian resource pada setiap cluster serta melakukan *scheduling*.
- Hadoop MapReduce sebagai programming model untuk memproses data dalam jumlah besar.

2.3 Hadoop Distributed File System (HDFS)

HDFS (*Hadoop Distributed File System*) adalah sistem file berbasis Java yang didistribusikan langsung di Hadoop. Sebagai sistem file terdistribusi, HDFS digunakan untuk memproses data berukuran besar karena memiliki ukuran blok yang lebih besar dibandingkan sistem file lainnya [16].

HDFS menyimpan file besar (biasanya dalam ukuran gigabyte hingga terabyte) pada beberapa mesin. HDFS akan melakukan proses pemecahan file besar menjadi bagian-bagian kecil, kemudian akan didistribusikan ke seluruh cluster komputer [17]. Adapun Gambar 3. di bawah ini merupakan blok diagram dari HDFS.



Gambar 3. Blok Diagram HDFS

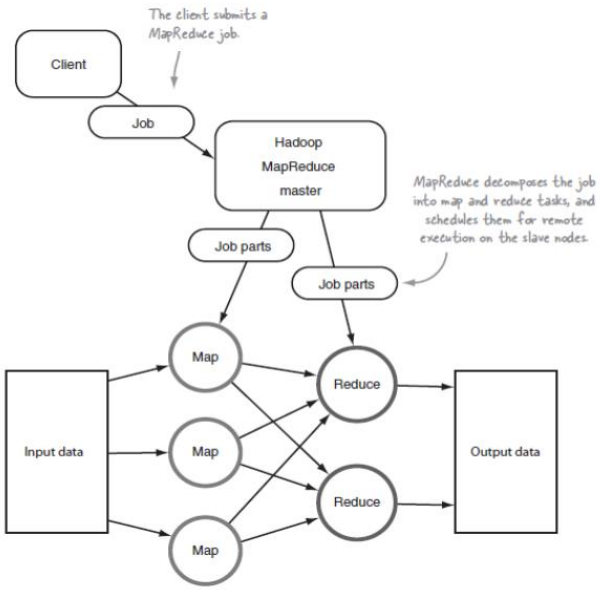
Pada Gambar 3 di atas, data dipecah menjadi bagian-bagian kecil yang disebut blok dan ukuran blok dapat diubah sesuai kebutuhan. Data atau file apa pun yang disimpan di HDFS akan selalu memiliki lebih dari satu salinan data atau file tersebut. Ini dinamakan *Factor Replication* (RF) dimana file disimpan dalam 3 DataNode maka jika salah satu DataNode rusak maka DataNode yang lain dapat menyediakan file tersebut [17].

HDFS juga memiliki dua jenis node yang beroperasi dalam arsitektur *master-slave* yaitu sebuah namenode (*master*) dan beberapa datanodes (*slaves*)[2]. NameNode bertanggung jawab untuk menyimpan informasi tentang penempatan blok-blok data di cluster Hadoop. Ia bertanggung jawab untuk mengatur dan mengendalikan blok-blok data tersimpan yang didistribusikan ke seluruh komputer yang membentuk cluster Hadoop. adapun, DataNode bertanggung jawab untuk menyimpan blok-blok data yang dialamatkan padanya dan melaporkannya ke NameNode secara berkala [11].

2.4 MapReduce

MapReduce adalah pemrograman paralel yang ada pada Hadoop yang berguna untuk memproses data dalam jumlah besar secara real time antar setiap node [18]. Hadoop MapReduce disediakan untuk menulis aplikasi yang memproses dan menganalisis kumpulan data besar secara parallel pada cluster multinode besar pada perangkat keras komoditas dengan cara yang dapat diskalakan, andal, dan toleran terhadap kesalahan [19]. MapReduce terdiri dari konsep fungsi Map dan Reduce yang biasa digunakan pada *functional programming* [20].

Model MapReduce adalah menyederhanakan pemrosesan dengan meringkas kompleksitas dalam bekerja dengan sistem terdistribusi seperti komputasi paralel, pembagian kerja, dan mengelola perangkat keras dan perangkat lunak yang tidak dapat diandalkan. Dengan abstraksi ini, MapReduce memungkinkan *programmer* untuk fokus pada kebutuhan bisnis, daripada menangani masalah sistem terdistribusi [2]. Untuk model kerja dari algoritma mapreduce dapat dilihat pada Gambar 4.



Gambar 4. Model Algoritma MapReduce

Pada saat mengolah data, Hadoop terlebih dahulu melakukan proses *mapping* pada task yang terdapat pada slot map hingga selesai, kemudian dilanjutkan dengan proses *reduce* pada *slot reduce*. MapReduce terdiri dari tiga tahap, yaitu tahap *map*, tahap *shuffle*, dan terakhir tahap *reduce*. Untuk tahapan shuffle dan reduce digabungkan ke dalam satu tahap besaran-nya yaitu tahap reduce [7]. Dimana :

- Operasi *map* adalah dengan membagi file input 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, misalnya file berbasis teks bergaris dibagi kedalam beberapa bagian dan dipecah secara *byte*. Lalu *map task* akan dibuat per *filesplit* yang ada.
- proses *reduce* adalah 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 [2].

2.5 Dataset Nasabah Bank

Bank adalah entitas bisnis yang mengelola dan menyimpan data besar dalam *database*, yang kemudian memprosesnya unutk menghasilkan sebuah informasi yang saling terkait tentang nasabah. Data ini dapat digunakan untuk memelihara hubungan antar bank dengan nasabah yang valid, dan juag berguna untuk menentukan penaawaran produk perbankan secara individual [21]. Dataset nasabah bank sendiri adalah kumpulan data yang berisi informasi tentang nasabah atau pelanggan sebuah bank. Dataset ini dapat berisi informasi yang relavan dengan aktivitas perbankan seperti identitas nasabah, informasi keuangan, aktivitas perbankan dan lain-lain.

3. METODELOGI PENELITIAN

Penelitian kali Ini akan mengikuti pendekatan metodologi yang komprehensif untuk menginvestigasi implementasi Hadoop Framework dalam analisis data nasabah bank. Metodologi ini akan memberikan panduan komprehensif untuk menjalankan penelitian ini dengan fokus pada implementasi Hadoop Framework dalam analisis data nasabah bank. Dengan mengikuti langkah-langkah ini, diharapkan penelitian ini akan memberikan kontribusi yang berharga untuk pemahaman kita tentang manfaat dan tantangan dalam menerapkan teknologi Big Data dalam sektor perbankan. Berikut adalah langkah-langkah metodologi yang akan diambil dalam penelitian ini :

Gambar 5. Tahap Penelitian

1. Studi Literatur

Melakukan tinjauan literatur yang komprehensif untuk memahami dasar-dasar Hadoop Framework. Menganalisis studi kasus, penelitian terkait, dan publikasi terkini untuk mengidentifikasi tren dan praktik terbaik dalam implementasi Hadoop. Untuk studi kasus yang kami angkat dalam percobaan ini adalah untuk mengkluster data nasabah bank unutk menentukan banyak transaksi setiap bulannya. Adapun dataset yang digunakan berasal dari Kaggle. Berikut tampilan dari dataset nasabah bank yang kami gunakan.

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
0	59	admin.	married	secondary	no	2343	yes	no	unknown	5	may	1042	1	-1	0	unknown	yes
1	56	admin.	married	secondary	no	45	no	no	unknown	5	may	1467	1	-1	0	unknown	yes
2	41	technician	married	secondary	no	1270	yes	no	unknown	5	may	1389	1	-1	0	unknown	yes
3	55	services	married	secondary	no	2476	yes	no	unknown	5	may	579	1	-1	0	unknown	yes
4	54	admin.	married	tertiary	no	184	no	no	unknown	5	may	673	2	-1	0	unknown	yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11157	33	blue-collar	single	primary	no	1	yes	no	cellular	20	apr	257	1	-1	0	unknown	no
11158	39	services	married	secondary	no	733	no	no	unknown	16	jun	83	4	-1	0	unknown	no
11159	32	technician	single	secondary	no	29	no	no	cellular	19	aug	156	2	-1	0	unknown	no
11160	43	technician	married	secondary	no	0	no	yes	cellular	8	may	9	2	172	5	failure	no
11161	34	technician	married	secondary	no	0	no	no	cellular	9	jul	628	1	-1	0	unknown	no

11162 rows × 17 columns

Gambar 6. Dataset Nasabah Bank

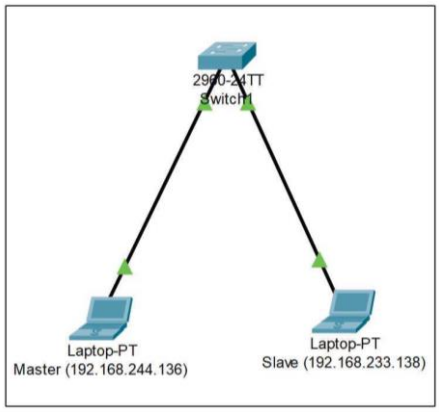
2. Perencanaan dan Perancangan Sistem

Sebelum mengimplementasikan Hadoop Framework, hal yang harus dilakukan terlebih dahulu adalah adalah melakukan perencanaan seperti Langkah-langkah apa saja yang perlu dilakukan agar implementasi Hadoop framework untuk analisis data nasabah bank dapat berjalan dengan efisien. langkah-langkah ini seperti menginstal hadoop dan apache Hive sebagai lingkungan big data penelitian pada masternode dan slavenode, melakukan konfigurasi untuk tiap node (bashrc, namenode, datanode, dll) agar dapat saling berinteraksi dan dengan bantuan java. Setelah kedua langkah tersebut dilakukan dengan benar, maka Hadoop, HDFS,

dan hive bisa dijalankan.ketika semua langkah tersebut telah dilakukan dengan benar dan sesuai maka barulah Hadoop, HDFS, dan hive bisa dijalankan untuk melakukan analisis big data. Untuk lebih jelasnya berikut alur perencanaan sebelum melakukan analisis big data.

Gambar 7. Alur Tahap Perencanaan

Dalam perancangan sistem hal pertama yang harus diperhatikan adalah topologi jaringan dari Hadoop. Berikut topologi jaringan pada arsitektur Hadoop.



Gambar 8. Topologi Jaringan

Pada gambar tersebut, kami membuat sebuah Topologi , dimana digunakan dua buah siatem operasi dengan yang satu menjadi master dan yang lainnya menjadi slave yang dihubungkan ke switch pada masing-masing sistem operasi. Topologi implementasi Hadoop adalah konfigurasi fisik atau logis dari komponen-komponen perangkat lunak dan perangkat keras dalam suatu kluster Hadoop. Ini merujuk pada cara komputer, server, dan perangkat penyimpanan terhubung dan diatur untuk menjalankan platform Hadoop guna mengelola dan menganalisis data besar.

3. Implementasi Hadoop Framework

Mengimplementasikan Hadoop Framework sesuai dengan perancangan sistem yang telah disiapkan. Untuk mengiplementasikan Apache Hadoop kami menggunakan VMware Workstation. VMware Workstation ini memungkinkan pengguna untuk membuat satu atau lebih mesin virtual dan menjalankannya secara bersamaan. Workstation sering digunakan untuk menguji sistem operasi baru, menjalankan aplikasi, atau menguji dampak virus tanpa

harus khawatir kehilangan data yang tersimpan di perangkat komputasi. Di mana kami menginstall sistem operasi Ubuntu pada VMware Workstation untuk dapat menginstall Apache Hadoop.

4. Analisis Data

Sebelum melakukan analisis hal pertama yang harus dilakukan adalah menginput dataset ke direktori HDFS. Setelah itu dataset akan diolah untuk melihat jumlah transaksi yang terjadi setiap bulannya menggunakan alat dan teknik analisis data yang sesuai, disini kami menggunakan metode MapReduce untuk mengolah dataset.

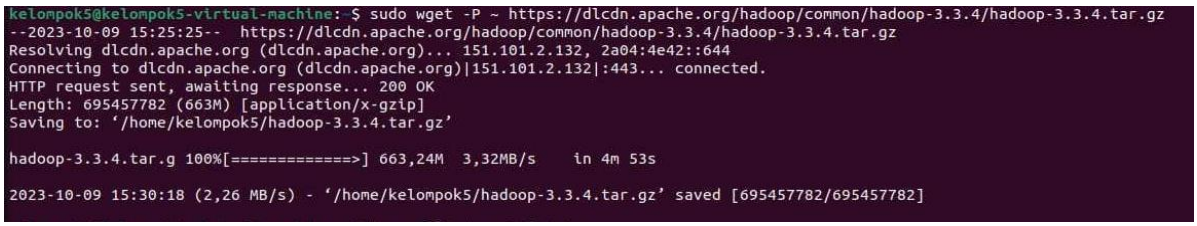
5. Kesimpulan

Dari semua tahapan yang dilakukan peneliti dalam penelitian ini, tahapan terakhir yang dilakukan yaitu menarik kesimpulan dari semua tahapan yang telah dilakukan.

4. HASIL PENELITIAN

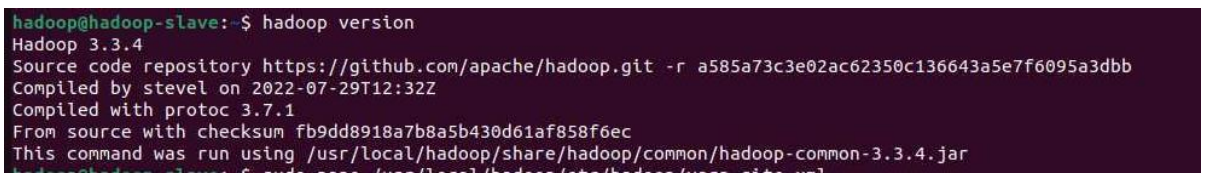
4.1 Menginstal Hadoop

Langkah pertama yang harus dilakukan dalam mengolah data pada Hadoop tentunya adalah menginstall apache Hadoop terlebih dahulu pada sistem operasi. Adaapun untuk melakukan penginstallan Hadoop pada sebuah sistem operasi contohnya ubuntu dapat dilakukan dengan mengetikkan perintah sudo wget -P ~ https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz. Setelah mengetikkan perintah tersebut pada terminal maka kita tinggal menunggu hingga proses instalasi Hadoop selesai. Berikut tampilan saat melakukan installasi Hadoop.



Gambar 9. Installasi Hadoop

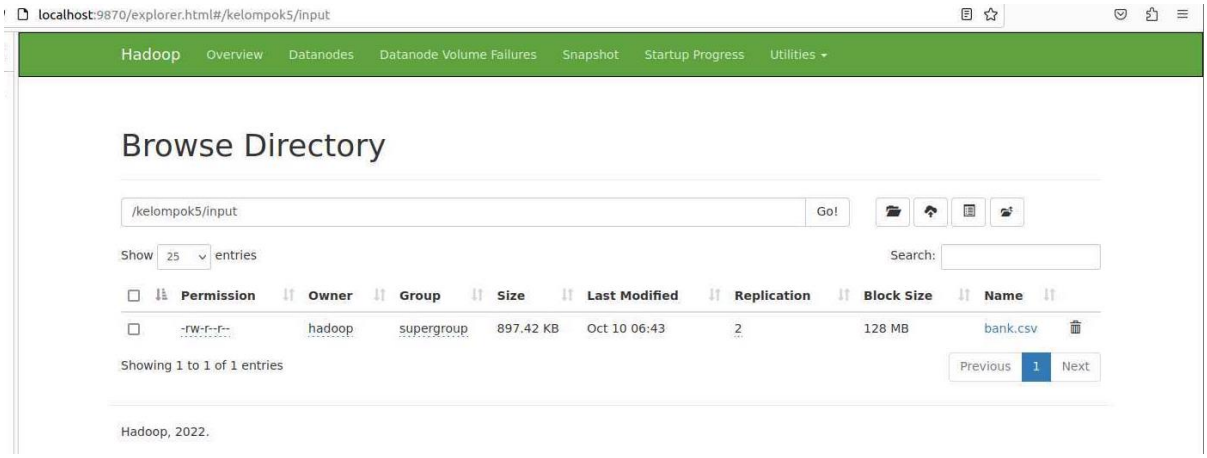
Unutk mengecek apakah Hadoop telah terinstall pada sistem operasi kita adalah dengan cara mengetikkan perintah Hadoop version pada terminal seperti pada Gambar 10. di bawah ini.



Gambar 10. Pengecekan Installasi Hadoop

4.2 Membuat dan Menginput Dataset Pada Direktori HDFS

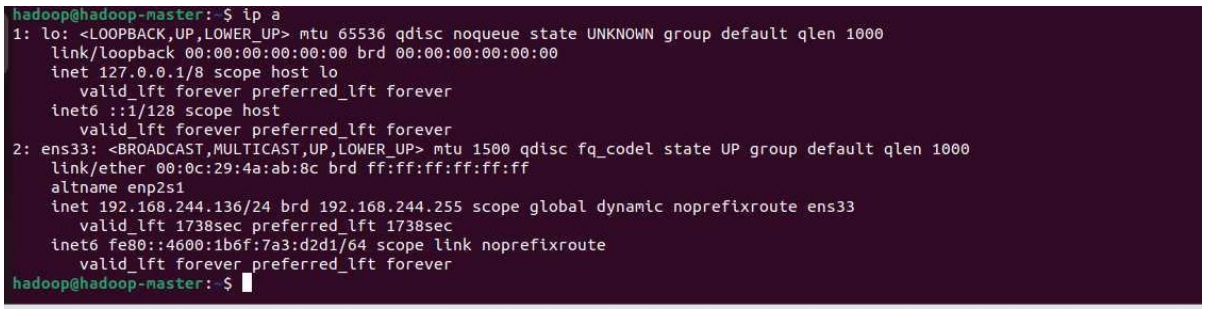
Sebelum menginput dataset, buat terlebih dahulu direktori baru pada HDFS sebagai tempat dataset tersebut disimpan. Untuk membuat direktori baru dapat dilakukan dengan mengetikkan perintah di command promt atau melalui browse directory yang bis akita akses melalui browser dengan menngetikkan "localhost:9870" pada kolom search. Di bawah ini merupakan tampilan Browse Directory pada HDFS.



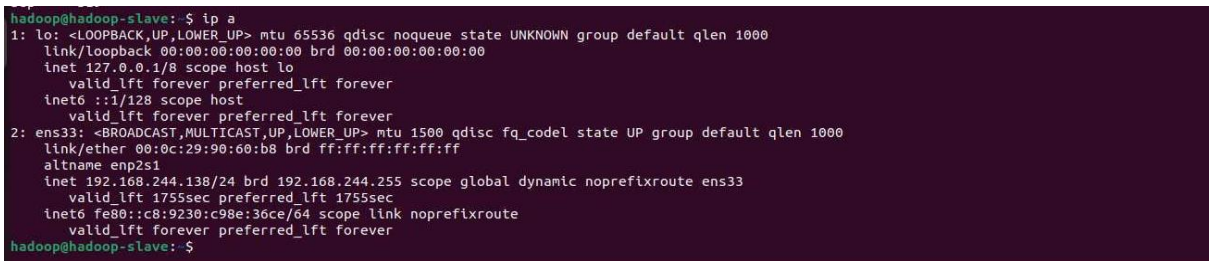
Gambar 11. Direktori Pada HDFS

4.3 Mengecek Ip Address

Selanjutnya kita harus memastikan bahwa kedua mesin telah berada pada jaringan yang sama. Adapun Untuk mengecek Ip address pada Hadoop-master dan Hadoop-slave dapat dilakukan dengan menjalankan perintah ip a pada terminal Hadoop-master dan Hadoop-slave seperti pada Gambar di bawah ini.



Gambar 12. Ip Address Pada Hadoop-Master

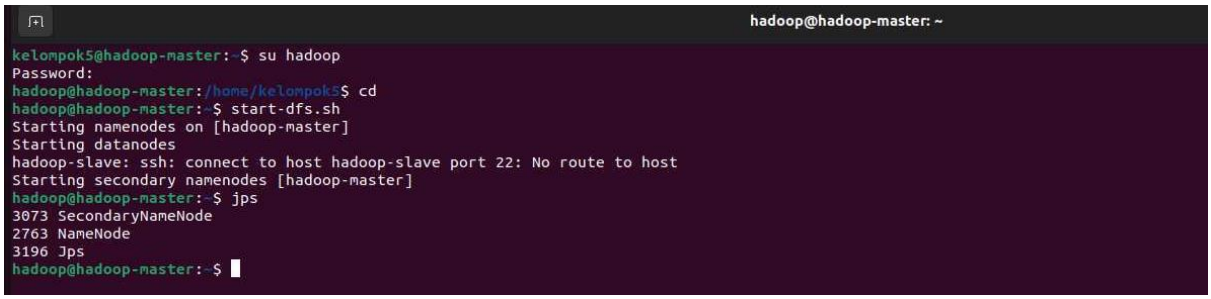


Gambar 13. Ip Address Pada Hadoop-Slave

Pada kedua gambar diatas terlihat bahwa kedua mesin telah berada pada jaringan yang sama. Di mana Ip address dari Hadoop-master adalah 192.168.244.136 dan pada Hadoop-slave Ip addressnya dalah 192.168.244.138.

4.4 Menghubungkan Hadoop-Master dengan Hadoop-Slave

Untuk menghubungkan Hadoop-Master dengan Hadoop-Slave dapat dilakukan dengan memulai HDFS. Adapun untuk memulai HDFS dapat dilakukan dengan menjalankan perintah start-dfs.sh seperti pada gambar di bawah ini.



Gambar 14. Menghubungkan NameNode

Gambar 15. Menghubungkan DataNode

Adapun untuk melihat apakah Hadoop-Slave sudah bisa mengakses Hadoop-Master, dapat dilihat pada Datanode usage histogram seperti pada Gambar 16. di bawah ini.

Gambar 16. DataNode HDFS Diakses Dengan Hadoop-Slave

Berdasarkan gambar diatas terlihat bahwa Hadoop-master dan Hadoop-slave telah terhubung. Hal ini dapat terlihat pada Gambar 16. dimana DataNode pada HDFS dapat diakses oleh Hadoop-slave. DataNode bertanggung jawab untuk menyimpan dan mengelola data fisik dari file yang disimpan dalam HDFS. DataNode berperan sebagai "slave" dalam arsitektur HDFS dan berfungsi untuk menyimpan sebagian data yang ada dalam kluster Hadoop.

4.5 Menjalankan MapReduce

Untuk melakukan cluster menggunakan metode MapReduce, pertama-tama kita harus membuat kode program map dan reduce terlebih dahuilu. Kode map merupakan kode yang melakukan pemetaan data input dalam format CSV menjadi pasangan kunci-nilai, dengan kunci berupa nama negara ("month") dan nilai 1. Hal ini digunakan dalam operasi MapReduce untuk menghitung jumlah kemunculan setiap negara dalam data input yang diberikan.

Sedangkan Kode reduce merupakan kode yang melakukan reduksi data dengan menggabungkan semua nilai yang memiliki kunci (month) yang sama, sehingga menghasilkan keluaran yang mencakup jumlah kemunculan setiap negara dalam data input yang diberikan. Gambar di bawah ini merupakan contoh kode program Map dan Reduce.

map.py

```
#!/usr/bin/env python
import sys
import csv

# Membaca baris-baris dari input
for line in sys.stdin:
    try:
        # Menghapus karakter newline dari baris
        line = line.strip()

        # Membuat pembaca CSV
        reader = csv.reader([line])

        # Membaca baris CSV
        for row in reader:
            # Pastikan baris memiliki setidaknya 11 kolom (indeks 10 adalah kolom "month")
            if len(row) > 10:
                # Ambil kolom "month"
                month = row[10]

                # Output kolom "month" sebagai kunci dan jumlah 1 sebagai nilai
                print(f"{month}\t1")
    except Exception as e:
        # Tangani kesalahan jika ada
        print(f"Error: {e}", file=sys.stderr)
```

Gambar 17. Kode Program Map

reduce.py

```
#!/usr/bin/env python
import sys

current_month = None
current_count = 0

# Membaca keluaran dari Mapper
for line in sys.stdin:
    line = line.strip()
    try:
        month, count = line.split("\t")

        # Menghitung total kemunculan bulan
        if current_month == month:
            current_count += int(count)
        else:
            if current_month:
                # Output hasil reduksi
                print(f"{current_month}\t{current_count}")
            current_month = month
            current_count = int(count)
    except ValueError:
        # Baris tidak dapat dibagi menjadi dua bagian, lewati
        continue

# Output terakhir
if current_month:
    # Output hasil reduksi terakhir
    print(f"{current_month}\t{current_count}")
```

Gambar 18. Kode Program Reduce

Selanjutnya unutk menjalankan MapReduce disini kami menggunakan Hadoop Streaming. Adapun unutk perintah yang digunakan dapat dilihat pada Gambar 19. di bawah ini.

```
hadoop@hadoop-slave:~$ hadoop jar /home/hadoop/hadoop/hadoop-streaming-3.3.4.jar -mapper "python3 /home/hadoop/hadoop/map.py" -reducer "python3 /home/hadoop/hadoop/reduce.py" -input
 /kelompok5/input/bank.csv -output /kelompok5/output1
2023-10-10 07:24:08,260 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-10-10 07:24:08,653 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-10-10 07:24:08,653 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-10-10 07:24:08,713 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2023-10-10 07:24:09,223 INFO mapred.FileInputFormat: Total input files to process : 1
2023-10-10 07:24:09,345 INFO mapreduce.JobSubmitter: number of splits:1
2023-10-10 07:24:09,771 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1443529916_0001
2023-10-10 07:24:09,771 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-10-10 07:24:09,949 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-10-10 07:24:09,951 INFO mapreduce.Job: Running job: job_local1443529916_0001
2023-10-10 07:24:09,962 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-10-10 07:24:09,964 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2023-10-10 07:24:09,994 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-10-10 07:24:10,001 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
```

Gambar 19. Menjalankan MapReduce

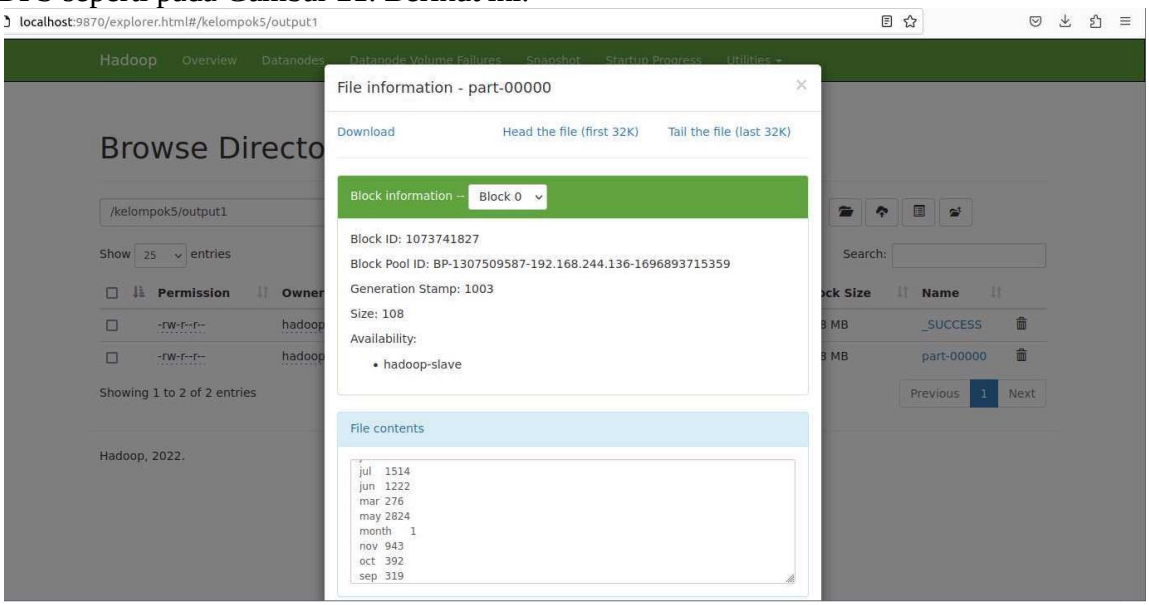
4.6 Hasil Penelitian

Untuk mengecek output dari hasil cluster MapReduce dapat dengan menggunakan perintah hdfs dfs –cat /kelompok5/output1/part-00000 pada Gambar 20. Di bawah ini.

```
hadoop@hadoop-slave:~$ hdfs dfs -cat /kelompok5/output1/part-00000
apr     923
aug     1519
dec     110
feb     776
jan     344
jul     1514
jun     1222
mar     276
may     2824
month   1
nov     943
oct     392
sep     319
```

Gambar 20. Output MapReduce

Selain itu, kita juga dapat melihat output cluster MapReduce pada browse directory di HDFS seperti pada Gambar 21. Berikut ini.



Gambar 21. Output MapReduce Pada HDFS Yang Diakses Hadoop-Slave

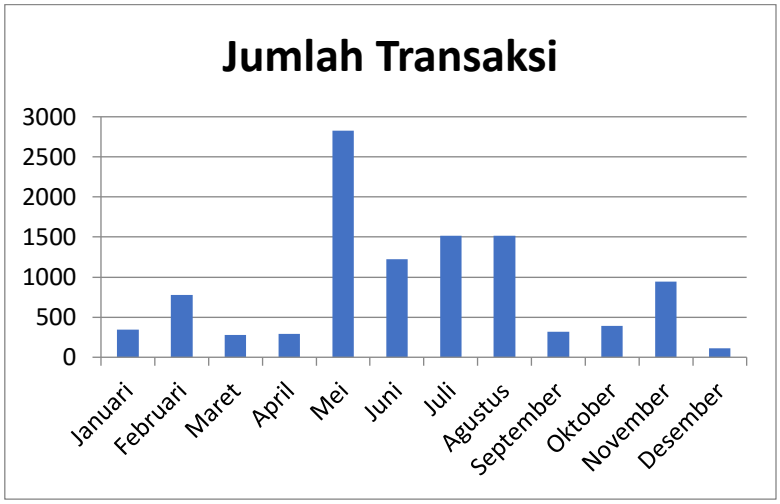
Dari Gambar 20. dan Gambar 21. di atas kita dapat mengetahui berapa jumlah transaksi yang terjadi dalam setiap bulan. Untuk outputnya seacar lebih jelas dapat dilihat pada Tabel 1. berikut.

Bulan	Jumlah Transaksi
Januari	344
Februari	776
Maret	276

April	293
Mei	2824
Juni	1222
Juli	1514
Agustus	1519
September	319
Oktober	392
November	943
Desember	110

Tabel 1. Tabel Hasil Percobaan

Adapun berikut grafik output jumlah transaksi nasabah bank.



Gambar 22. Grafik Hasil Percobaan

Berdasarkan output yang dihasilkan dapat disimpulkan bahwa transaksi yang paling banyak adalah pada bulan Mei yaitu dengan transaksi sebanyak 2.824 kali dan transaksi yang paling sedikit adalah pada bulan Desember yaitu dengan transaksi sebanyak 110 kali.

5. KESIMPULAN

Hadoop adalah sebuah kerangka kerja untuk pemrosesan dan penyimpanan data dalam skala besar atau disebut big data. Hadoop juga terdiri atas banyak modul salah satunya Hadoop Distribute File System (HDFS) MapReduce. Dengan HDFS kita dapat memproses dan menyimpan data dalam skala besar. Pada HDFS ini data akan dibagi menjadi blok-blok kecil dan didistribusikan di berbagai node dalam cluster seperti NameNode dan DataNode. NameNode disini akan di atur metadata dari dataset seperti dtruktur direktori, izin, dan informasi lainnya. Selanjutnya pada DataNode, disinilah data actual disimpan dalam bentuk blok-blok.

Dalam percobaan ini, digunakan pemrosesan secara multimode sehingga tugas-tugas dapat dilakukan secara paralel di banyak node. Hal ini menghasilkan kinerja yang lebih tinggi karena pemrosesan data dibagi menjadi tugas-tugas kecil yang dapat dijalankan secara bersamaan. Adapun untuk memudahkan dalam menganalisis data besar dalam paralel di seluruh kluster maka digunakan metode MapReduce.

Hasil akhir dari percobaan ini adalah output yang dihasilkan dari proses MapReduce dimana hasil yang diperoleh adalah jumlah transaksi yang terjadi pada setiap bulannya. Berdasarkan output yang didapat, dapat disimpulkan bahwa transaksi yang paling banyak terjadi adalah pada bulan Mei yaitu dengan transaksi sebanyak 2.824 kali dan transaksi yang paling sedikit adalah pada bulan Desember yaitu dengan transaksi sebanyak 110 kali. Dengan begitu cluster Mapreduce ini dapat digunakan untuk menganalisis tren, memprediksi perilaku masa depan, dan membuat proyeksi bisnis yang lebih baik. Misalnya untuk melihat bulan-bulan dengan peningkatan atau penurunan signifikan dalam jumlah transaksi.

DAFTAR PUSTAKA

[1] I. R. Prabaswara and R. Saputra, "Analisis Data Sosial Media Twitter Menggunakan Hadoop dan Spark," *IT JOURNAL RESEARCH AND DEVELOPMENT*, vol. 4, no. 2, Mar. 2020, doi: 10.25299/itjrd.2020.vol4(2).4099.

[2] K. Basuki, H. Novianus Palit, and L. P. Dewi, "Implementasi Hadoop: Studi Kasus Pengolahan Data Peminjaman Perpustakaan Universitas Kristen Petra."

[3] N. Fajriyah *et al.*, "IMPLEMENTASI TEKNOLOGI BIG DATA DI ERA DIGITAL".

[4] Icha Yasin, "Data Besar, Data Analisis, dan Pengembangan Kompetensi Pustakawan".

[5] B. Maryanto, "BIG DATA DAN PEMANFAATANNYA DALAM BERBAGAI SEKTOR," 2017.

[6] M. Hana and J. Marzal, "PEMBANGUNAN INFRASTURKTUR BIG DATA BERBASIS HADOOP PADA UNIVERSITAS JAMBI," 2018.

[7] R. Adawiyah and S. Munir, "Jurnal Informatika Terpadu ANALISIS DAN EVALUASI ALGORITMA MAPREDUCE WORDCOUNT PADA CLUSTER HADOOP MENGGUNAKAN INDIKATOR KECEPATAN," *Jurnal Informatika Terpadu*, vol. 6, no. 1, pp. 14–19, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[8] C. Wibawa, S. Wirawan, M. Mustikasari, and D. T. Anggraeni, "KOMPARASI KECEPATAN HADOOP MAPREDUCE DAN APACHE SPARK DALAM MENGOLAH DATA TEKS," *Jurnal Ilmiah MATRIK*, vol. 24, no. 1, 2022.

[9] M. Awaluddin, R. Angelia Mahlil, and L. Ode Muhammad Saidi, "Implementasi Hadoop Mapreduce Untuk Memprediksi Predikat Kelulusan Mahasiswa," *Journal on Education*, vol. 05, no. 04, pp. 17239–17251, 2023.

[10] S. R. M. Zeebaree, H. M. Shukur, L. M. Haji, R. R. Zebari, K. Jacksi, and S. M. Abas, "Characteristics and Analysis of Hadoop Distributed Systems."

[11] S. Oliviandi, A. B. Osmond, and R. Latuconsina, "IMPLEMENTASI APACHE SPARK PADA BIG DATA BERBASIS HADOOP DISTRIBUTED FILE SYSTEM IMPLEMENTATION APACHE SPARK ON BIG DATA BASED HADOOP DISTRIBUTED FILE SYSTEM."

[12] E. E. Supriyanto, I. Susilo Bakti, M. Furqon, and R. Artikel, "THE ROLE OF BIG DATA IN THE IMPLEMENTATION OF DISTANCE LEARNING INFO ARTIKEL ABSTRAK," vol. 12, no. 1, pp. 61–68, 2021, doi: 10.31764.

[13] H. K. Nafah1 and E. Purnaningrum2, "Seminar Nasional Hasil Riset dan Pengabdian Ke-III (SNHRP-III 2021) PENGGUNAAN BIG DATA MELALUI ANALISIS GOOGLE TRENDS UNTUK MENGETAHUI PERSPEKTIF PARIWISATA INDONESIA DI MATA DUNIA." [Online]. Available: https://trends.google.com/trends/?geo=US

[14] R. Sunartio, H. Novianus Palit, A. Gunawan, and K. Kunci, "Hotel Recommender System Menggunakan Metode Pendekatan Graph pada Dataset Trivago."

[15] P. Grover and A. K. Kar, "Big Data Analytics: A Review on Theoretical Contributions and Tools Used in Literature," *Global Journal of Flexible Systems Management*, vol. 18, no. 3, pp. 203–229, Sep. 2017, doi: 10.1007/s40171-017-0159-3.

[16] M. I. Syafi'i, A. Bhawiyuga, and M. Data, "Analisis Perbandingan Kinerja File System GlusterFS dan HDFS dengan Skenario Distribusi Striped dan Replicated," 2019. [Online]. Available: http://j-ptiik.ub.ac.id

[17] Y. Surahman and H. Saptono, "Jurnal Informatika Terpadu EVALUASI KINERJA HDFS SEBAGAI INFRASTRUKTUR PEMBANGUNAN BIG DATA," *Jurnal Informatika Terpadu*, vol. 4, no. 2, pp. 63–70, 2018, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[18] H. M. Putra, T. Akbar, A. Ahmadi, and M. I. Darmawan, "Analisa Performa Klastering Data Besar pada Hadoop," *Infotek : Jurnal Informatika dan Teknologi*, vol. 4, no. 2, pp. 174–183, Jul. 2021, doi: 10.29408/jit.v4i2.3565.

[19] M. R. Ghazi and D. Gangodkar, "Hadoop, mapreduce and HDFS: A developers perspective," in *Procedia Computer Science*, Elsevier B.V., 2015, pp. 45–50. doi: 10.1016/j.procs.2015.04.108.

[20] W. Saputra *et al.*, "ANALISIS KINERJA ALGORITMA DELAY SCHEDULING PADA HADOOP TERHADAP KARAKTERISTIK RESPONS TIME UNTUK PENGIRIMAN 2 JOB YANG BERBEDA," vol. 5, no. 1, 2020.

[21] P. Subarkah, E. P. Pambudi, and S. O. N. Hidayah, "Perbandingan Metode Klasifikasi Data Mining untuk Nasabah Bank Telemarketing," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 1, pp. 139–148, Sep. 2020, doi: 10.30812/matrik.v20i1.826.