

TESIS

**ANALISIS PEMANFAATAN TEKNOLOGI BLOCKCHAIN
DAN INTERPLANETARY FILE SYSTEM (IPFS) UNTUK
MENGATASI SINGLE POINT OF FAILURE PADA HADOOP**

*Analysis of Blockchain and Interplanetary File System (IPFS) Utilization
for Overcome Single Point of Failure in Hadoop*

ASRUL PAELORI AHMAD

D082201008



**PROGRAM STUDI S2 TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**



Optimized using
trial version
www.balesio.com

TESIS

ANALISIS PEMANFAATAN TEKNOLOGI BLOCKCHAIN DAN INTERPLANETARY FILE SYSTEM (IPFS) UNTUK MENGATASI SINGLE POINT OF FAILURE PADA HADOOP

ASRUL PAELORI AHMAD
D082201008

Telah dipertahankan di hadapan Panitia Ujian Tesis yang dibentuk dalam rangka penyelesaian studi
pada Program Magister Teknik Informatika Fakultas Teknik

Universitas Hasanuddin

Pada tanggal 23 Februari 2024

dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama



Dr. Eng. Ady Wahyudi Paundu, ST., MT.
NIP. 19840403 201012 1 004

Pembimbing Pendamping



Dr. Amil Ahmad Ilham, S.T., M.IT
NIP. 19731010 199802 1 001

Dekan Fakultas Teknik
Universitas Hasanuddin



mmad Isran Ramli, M.T. IPM., ASEAN.Eng.
NIP. 19730926 200012 1 002

Ketua Program Studi
S2 Teknik Informatika



Dr. Ir. Zahir Zainuddin, M.Sc.
NIP. 19640427 198910 1 002



PENGAJUAN TESIS

ANALISIS PEMANFAATAN TEKNOLOGI BLOKCHCHAIN DAN INTERPLANETARY FILE SYSTEM (IPFS) UNTUK MENGATASI SINGLE POINT OF FAILURE PADA HADOOP

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi Teknik Informatika

Disusun dan diajukan oleh

Asrul Paelori Ahmad

D082201008

Kepada

**FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**



PERNYATAAN KEASLIAN TESIS DAN PELIMPAHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

Nama : Asrul Paelori Ahmad
Nomor Mahasiswa : D082201008
Program Studi : S2 Teknik Informatika

Dengan ini menyatakan bahwa, tesis berjudul “Analisis Pemanfaatan Teknologi Blockchain dan Interplanetary File System (IPFS) untuk Mengatasi Single Point Of Failure pada Hadoop” adalah karya saya dengan arahan dari komisi pembimbing (Dr. Eng. Ady Wahyudi Paundu, ST., M.T. dan Dr. Amil Ahmad Ilham, ST., M.IT). Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apapun kepada perguruan tinggi manapun. Sumber informasi yang berasal 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 isi tesis ini telah dipublikasikan di Conference : The 2023 IEEE International Conference on Communication, Networks, and Satellite (ComNetSat) sebagai artikel dengan judul “Analysis of Blockchain and Interplanetary File System (IPFS) Utilization for Big Data Architecture Optimization”.

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

Gowa, Februari 2024

Yang menyatakan



METERAI
TEMPEL
43ALX129014068

Asrul Paelori Ahmad



KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul “ANALISIS PEMANFAATAN TEKNOLOGI BLOCKCHAIN DAN INTERPLANETARY FILE SYSTEM (IPFS) UNTUK MENGATASI SINGLE POINT OF FAILURE PADA HADOOP” 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 tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. 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 di setiap langkah dalam pembuatan program hingga penulisan laporan skripsi ini;
2. Kedua Orang tua penulis, Bapak H. Ahmad dan Ibu Hj. Nurhayati S, S.Pd; Istri penulis, Siswanty Baharuddin, S.Pt yang selalu memberikan dukungan, doa, dan semangat;
3. Bapak Dr. Eng. Ady Wahyudi Paundu, ST., M.T. selaku pembimbing I dan Bapak Dr. Amil Ahmad Ilham, S.T., M.IT. selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;
4. Bapak Adnan, S.T., M.T., Bapak Dr. Eng. Muhammad Niswar, ST., M.IT., dan Ibu Mukarramah Yusuf, B.Sc., M.Sc., Ph.D., selaku dosen penguji yang telah memberikan saran sehingga laporan tesis ini menjadi lebih baik;
5. Bapak Dr. Ir. Zahir Zainuddin, M.Sc selaku Ketua Departemen Magister Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;



ahabat, teman-teman dan kakak-kakak pascasarjana UNHAS yang telah erikan begitu banyak bantuan selama penelitian dan diskusi *progress* sunan tugas akhir;

7. Teman-teman pascasarjana UNHAS Angkatan 2 atas dukungan dan semangat yang diberikan selama ini;
8. Ibu Yuanita serta segenap Staf Departemen Teknik Informatika yang telah membantu penulis;
9. Orang-orang berpengaruh lainnya yang tidak sempat disebutkan oleh penulis.

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

Gowa, Februari 2024

Asrul Paelori Ahmad



ABSTRAK

ASRUL PAELORI AHMAD. *Analisis Pemanfaatan Teknologi Blockchain dan Inpterplanetary File System (IPFS) untuk Mengatasi Single Point of Failure pada Hadoop* (dibimbing oleh **Ady Wahyudi Paundu, Amil Ahmad Ilham**)

Big Data merupakan kumpulan data set yang sangat besar atau kompleks yang sangat susah untuk diolah oleh aplikasi pengolah data tradisional. Data ini bersumber dari berbagai perangkat atau media yang saling terkoneksi dengan internet. Dewasa ini pengolahan big data menggunakan Hadoop yang mampu memproses data dalam jumlah besar, namun sebagai paradigma penyimpanan terpusat Hadoop rentan terhadap Single Point of Failure, ketersediaan yang rendah, kegagalan sistem atau bahkan serangan hacker. Pada penelitian ini rancang sebuah arsitektur big data yang menggunakan Blockchain dan IPFS dan dilakukan perbandingan antara arsitektur Big Data menggunakan Hadoop dengan arsitektur Big Data menggunakan Blockchain dan Interplanetary File System (IPFS), serta melakukan pengujian terhadap waktu, throughput dan latency yang dibutuhkan masing-masing arsitektur. Hasil akhir dari penelitian ini adalah pemabahasan mengenai perbandingan eksekusi data berdasarkan waktu, latency dan throughput pada masing masing arstiktur. Dengan hasil perbandingan tersebut diperoleh bahwa throughput, latency dan waktu yang dihasilkan oleh arsitektur big data menggunakan Hadoop lebih baik dengan nilai rasio perbandingan untuk file size 500 MB yaitu 33, file size 700 MB yaitu 58, dan file size 1 GB yaitu 35.

Kata Kunci: Big Data, Blockchain, Interplanetary File System (IPFS), Cloud, Hadoop, Single Point of Failure, Throughput



ABSTRACT

ASRUL PAELORI AHMAD. *Analysis of Blockchain and Interplanetary File System (IPFS) Utilization for Overcome Single Point of Failure in Hadoop* (supervised by **Ady Wahyudi Paundu, Amil Ahmad Ilham**)

Big data is the collection of very large or complex data sets that are very difficult to process by traditional data processing applications. This data comes from various devices or media that are connected to the internet. Today, big data processing uses Hadoop, which are capable of processing large amounts of data. However, as centralized storage paradigms, Hadoop are vulnerable to Single Point of Failure, low availability, system failure, or even hacker attacks. In this research, a big data architecture using Blockchain and Interplanetary File System (IPFS) is designed and a comparison is made between big data architecture using Hadoop and big data architecture using Blockchain and IPFS and testing the time, throughput and latency required by each architecture. The final result of this research is a discussion of the comparison of data execution based on time, latency and throughput in each architecture. The comparison result shows that the throughput, latency and time generated by Hadoop's big data architecture are better with a comparison ratio value for file size 500 MB of 33, file size 700 MB of 58, and file size 1 GB of 35.

Kata Kunci: Big Data, Blockchain, Interplanetary File System (IPFS), Cloud, Hadoop, Single Point of Failure, Throughput



DAFTAR ISI

| | |
|--|------|
| HALAMAN JUDUL..... | i |
| LEMBAR PENGESAHAN PERSETUJUAN UJIAN | ii |
| PENGAJUAN TESIS..... | iii |
| PERNYATAAN KEASLIAN TESIS DAN PELIMPAHAN HAK CIPTA | iv |
| KATA PENGANTAR | v |
| ABSTRAK | vii |
| ABSTRACT..... | viii |
| DAFTAR ISI..... | ix |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR | xiii |
| DAFTAR GRAFIK..... | xv |
| DAFTAR LAMPIRAN | xvi |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Tujuan Penelitian..... | 3 |
| 1.4 Manfaat Penelitian..... | 4 |
| 1.5 Batasan Masalah..... | 4 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB II TINJAUAN PUSTAKA..... | 6 |
| 2.1 Big Data..... | 6 |
| Hadoop | 7 |
| Hadoop Distributed File System (HDFS) | 7 |
| Map Reduce | 9 |



| | | |
|---|---|-----------|
| 2.3 | Blockchain..... | 10 |
| 2.3.1. | Proof of Work Consensus Model..... | 12 |
| 2.3.2. | Proof of Stake Consensus Model..... | 13 |
| 2.4 | InterPlanetary File System (IPFS)..... | 14 |
| 2.5 | MetaMask..... | 16 |
| 2.6 | Smart Contract..... | 17 |
| 2.7 | Apache JMeter..... | 17 |
| 2.8 | Tinjauan Empiris | 18 |
| BAB III METODOLOGI PENELITIAN..... | | 26 |
| 3.1 | Tahap Penelitian | 26 |
| 3.2 | Waktu dan Lokasi Penelitian..... | 27 |
| 3.3 | Instrumen Penelitian..... | 27 |
| 3.4 | Tahap Persiapan..... | 28 |
| 3.5 | Kerangka Pemikiran | 28 |
| 3.6 | Sumber Data | 30 |
| 3.7 | Perancangan Arsitektur Big Data Hadoop | 30 |
| 3.8 | Perancangan Arsitektur Big Data Blockchain dan IPFS | 31 |
| 3.9 | Pengujian Sistem | 33 |
| 3.9.1 | Skenario Pengujian..... | 33 |
| 3.9.2 | Througput..... | 38 |
| 3.9.3 | Waktu | 39 |
| 3.9.4 | Latency | 39 |
| BAB IV HASIL DAN PEMBAHASAN | | 40 |
| 4.1 | Hasil Pengujian Arsitektur Hadoop..... | 40 |
| | Hasil Pengujian File Test 500 MB..... | 40 |
| | Hasil Pengujian File Test 700 MB..... | 41 |



| | | |
|---------------------------------|---|----|
| 4.1.3. | Hasil Pengujian File Test 1 GB..... | 42 |
| 4.2 | Hasil Pengujian Arsitektur Big Data Blockchain dan IPFS | 42 |
| 4.2.1. | Hasil Pengujian File Test 500 MB | 42 |
| 4.2.2 | Hasil Pengujian File Test 700 MB | 44 |
| 4.2.3 | Hasil Pengujian File Test 1 GB..... | 45 |
| 4.3 | Pengujian Throughput, Latency dan Waktu | 46 |
| 4.3.1 | Pengujian Arsitektur Big Data Hadoop..... | 46 |
| 4.3.2 | Pengujian Arsitektur Big Data Blockchain dan IPFS | 46 |
| 4.4 | Pembahasan | 46 |
| BAB V KESIMPULAN DAN SARAN..... | | 52 |
| 5.1. | Kesimpulan..... | 52 |
| 5.2. | Saran | 53 |
| DAFTAR PUSTAKA | | 54 |
| LAMPIRAN | | 57 |



DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Tabel Subsystem IPFS | 15 |
| Tabel 2. 2 Tabel State Of the Art | 19 |
| Tabel 3. 1 Tabel Pencapaian yang Diharapkan | 29 |
| Tabel 4. 1 Hash File dan Transaction Receipt File Test 500 MB | 43 |
| Tabel 4. 2 Hash File dan Transaction Receipt File Test 700 MB | 44 |
| Tabel 4. 3 Hash File dan Transaction Receipt File Test 1 GB..... | 45 |
| Tabel 4. 4 Nilai Throughput dan Waktu Pengujian Hadoop..... | 46 |
| Tabel 4. 5 Nilai Throughput dan Waktu Pengujian Blockchain dan IPFS | 46 |
| Tabel 4. 6 Perbandingan Nilai Rasio Throughput Hadoop dan IPFS | 49 |
| Tabel 4. 7 Perbandingan Nilai Pengujian Hadoop 3 dan 5 Node..... | 50 |
| Tabel 4. 8 Perbandingan Nilai Pengujian Blockchain dan IPFS 3 dan 5 Node .. | 50 |



DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Arsitektur HDFS | 8 |
| Gambar 2. 2 Alur MapReduce Untuk Kasus WordCount..... | 9 |
| Gambar 2. 3 Struktur Blockchain..... | 10 |
| Gambar 2. 4 Transaksi pada Blockchain..... | 11 |
| Gambar 2. 5 Perbandingan HTTPS dengan IPFS | 14 |
| | |
| Gambar 3. 1 Diagram Tahapan Penelitian | 26 |
| Gambar 3. 2 Alur Kerja Penelitian..... | 29 |
| Gambar 3. 3 Arsitektur Big Data Hadoop..... | 30 |
| Gambar 3. 4 Arsitektur Hadoop Multi Node | 31 |
| Gambar 3. 5 Arsitektur Big Data Blockchain dan IPFS | 31 |
| Gambar 3. 6 Integrasi Peers, IPFS dan Blockchain | 32 |
| Gambar 3. 7 Flowchart Pengujian Arsitektur Big Data Hadoop tanpa Single Point of Failure | 35 |
| Gambar 3. 8 Flowchart Pengujian Arsitektur Big Data Hadoop dengan Single Point of Failure..... | 35 |
| Gambar 3. 9 Hubungan Node pada Pengujian Arsitektur Big Data Hadoop..... | 36 |
| Gambar 3. 10 Flowchart Pengujian Arsitektur Big Data Blockchain dan IPFS tanpa Single Point of Failure..... | 37 |
| Gambar 3. 11 Flowchart Pengujian Arsitektur Big Data Blockchain dan IPFS dengan Single Point of Failure..... | 37 |
| Gambar 3. 12 Peer IPFS pada Pengujian Arsitektur Big Data Blockchain dan IPFS | 38 |
| Gambar 4. 1 Hasil Pengujian Single Point of Failure | 40 |
| Gambar 4. 2 Hasil Pengujian File Test 500 MB tanpa Single Point of Failure | 40 |
| Gambar 4. 3 Hasil Pengujian File Test 500 MB dengan Single Point of Failure . | 41 |
| Gambar 4. 4 Hasil Pengujian File Test 700 MB tanpa Single Point of Failure ... | 41 |
| Gambar 4. 5 Hasil Pengujian File Test 700 MB dengan Single Point of Failure | 41 |
| Gambar 4. 6 Hasil Pengujian File Test 1 GB tanpa Single Point of Failure | 42 |
| Gambar 4. 7 Hasil Pengujian File Test 1 GB dengan Single Point of Failure | 42 |



| | |
|---|----|
| Gambar 4. 8 Hasil Pengujian IPFS File Test 500 MB tanpa Single Point of Failure | 43 |
| Gambar 4. 9 Hasil Pengujian IPFS File Test 500 MB dengan Single Point of Failure | 43 |
| Gambar 4. 10 Hasil Pengujian IPFS File Test 700 MB tanpa Single Point of Failure | 44 |
| Gambar 4. 11 Hasil Pengujian IPFS File Test 700 MB dengan Single Point of Failure | 44 |
| Gambar 4. 12 Hasil Pengujian IPFS File Test 1 GB tanpa Single Point of Failure | 45 |
| Gambar 4. 13 Hasil Pengujian IPFS File Test 1 GB dengan Single Point of Failure | 45 |



DAFTAR GRAFIK

| | |
|---|----|
| Grafik 4. 1 Perbandingan Throughput tanpa Single Point of Failure | 47 |
| Grafik 4. 2 Perbandingan Throughput dengan Single Point of Failure..... | 47 |
| Grafik 4. 3 Perbandingan Waktu tanpa Single Point of Failure..... | 47 |
| Grafik 4. 4 Perbandingan Waktu dengan Single Point of Failure..... | 48 |
| Grafik 4. 5 Perbandingan Latency tanpa Single Point of Failure | 48 |
| Grafik 4. 6 Perbandingan Latency dengan Single Point of Failure..... | 48 |



DAFTAR LAMPIRAN

| | |
|---|----|
| Lampiran 1 Source code Smart Contract (storehash.sol)..... | 57 |
| Lampiran 2 Source code DApps (App.js) | 57 |
| Lampiran 3 Source code DApps (storehash.js)..... | 62 |
| Lampiran 4 Source code Koneksi ke IPFS (ipfs.js) | 63 |
| Lampiran 5 Source code Koneksi Web3 Metamask (web3.js)..... | 63 |



BAB I PENDAHULUAN

1.1 Latar Belakang

Big data merupakan kumpulan data dalam aplikasi yang begitu besar dan kompleks sehingga memerlukan penyimpanan data yang canggih, unik, manajemen, analisa dan teknologi visual. Big data adalah terminology dari data yang besar, bervariasi, data dihasilkan dengan cepat, dan memiliki kualitas dan keaslian, untuk menghasilkan sebuah nilai yang dikenal dengan 5V (Rajarshi Mitra, 2021).

Seiring dengan perkembangan jaman, berbagai perangkat dan media terkoneksi dengan internet menghasilkan jumlah data yang sangat besar. Sebagai contoh adalah Facebook memiliki lebih dari 300 Petabyte data yang berisi profil pribadi, gambar, video serta pesan pengguna. Lebih dari 2.5 quintillion bytes data dihasilkan setiap hari, dan diprediksikan setiap orang akan menghasilkan 1.7 MB data setiap detik (Rajarshi Mitra, 2021). Karena volume data yang sangat besar sehingga hampir tidak mungkin untuk sistem tradisional untuk memproses dan menganalisanya.

Pengelolaan big data yang berukuran besar, membuat teknik pemrosesan tradisional seperti Database Management System (DBMS) atau Relational Database Management System (RDBMS) tidak dapat digunakan lagi. Saat ini sistem platform yang dikenal untuk pengelolaan big data yaitu Hadoop. Hadoop mendukung untuk melakukan pengelolaan big data secara terdistribusi dengan melibatkan beberapa kluster komputer. Namun, sebagai paradigma penyimpanan terpusat, penyimpanan Hadoop rentan terhadap single point of failure, ketersediaan, kegagalan sistem, atau bahkan serangan hacking, yang menghasilkan banyak kasus pelanggaran data atau kehilangan data dan mekanisme yang rumit untuk share data antara pemangku kepentingan. Pada saat ini, potensi big data dalam memberikan kontribusi tidak dapat diabaikan.

Salah satu komponen utama pada Hadoop, yaitu Hadoop Distributed File System (HDFS) dan MapReduce. HDFS adalah media penyimpanan data pada Hadoop dan MapReduce adalah media pemrosesan data. Cara kerja Hadoop



mengadopsi prinsip Master dan Slave, master terdiri dari Namenode dan JobTracker, sedangkan slave terdiri dari Datanode dan TaskTracker. Namenode ini merupakan bagian yang dapat menyebabkan kegagalan fungsi (Single Point of Failure). Single Point of Failure keadaan dimana Namenode berhenti berfungsi yang mengakibatkan keseluruhan server pada kluster Hadoop berhenti bekerja dan tidak dapat diakses. Untuk mencegah resiko tersebut, arsitektur Hadoop 2 telah memfasilitasi fungsionalitas Namenode ganda, fitur ini mampu menyediakan fungsi Passive Standby Namenode untuk membackup Namenode yang aktif secara otomatis jika terjadi kegagalan fungsi.

Salah satu cara yang dapat dilakukan untuk mengatasi permasalahan Single Point of Failure adalah pemanfaatan teknologi Blockchain dan Interplanetary File System (IPFS). Teknologi blockchain pada dasarnya diciptakan untuk membantu dan mencatat transaksi tanpa otoritas terpusat, ini memungkinkan desentralisasi melalui teknologi buku besar, teknologi blockchain juga berjalan terus menerus tanpa down time sehingga memberikan ketersediaan yang tinggi. Sedangkan IPFS sendiri merupakan media penyimpanan dan pengelolaan file peer to peer yang sepenuhnya terdesentralisasi, dimana file direplikasi di semua node yang terhubung.

Beberapa penelitian terkait dengan pengoptimalan arsitektur Big Data antara lain (Kumar et al., 2017) penelitian ini memiliki beberapa kelebihan seperti membuat blockchain sebagai media penyimpanan metadata, sehingga menghilangkan peran Namenode pada arsitektur big data, sehingga menghasilkan penundaan akses metadata yang rendah dan meningkatkan waktu eksekusi permintaan data pada Hadoop. Namun dilain sisi penggunaan blockchain untuk menyimpan metadata menimbulkan permasalahan pada skalabilitas disebabkan karena blockchain tidak mampu untuk menyimpan dan memproses data dalam jumlah besar.

(Brisbane, 2016) Penelitian ini mengimplementasikan Hadoop dan penyimpanan terdesentralisasi yaaitu Interplanetary File System (IPFS). Hal ini cinkan tugas MapReduce dijalankan secara langsung pada daya yang 1 dan juga meningkatkan desentralisasi hosting data melalui replikasi



(Jhon *et al.*, 2021) Pada penelitian ini, dirumuskan model untuk penempatan, replikasi dan pemrosesan Big Data dengan menggabungkan fitur-fitur Hadoop dan IPFS. HDFS dan IPFS secara bersama-sama menangani tugas penempatan dan replikasi data dan kerangka kerja pemrograman MapReduce di Hadoop menangani tugas pemrosesan data. Hasilnya menunjukkan bahwa kerangka kerja ini mampu mencapai penyimpanan yang hemat biaya serta pemrosesan data yang lebih cepat.

Penelitian yang dilakukan akan memanfaatkan Interplanetary File System (IPFS) untuk menyimpan data dan Blockchain untuk menyimpan hash data yang dihasilkan oleh IPFS. Penelitian ini diharapkan mampu untuk mengatasi permasalahan Single Point of Failure (SPoF) dan selain itu diharapkan pula dapat menjadi salahsatu alternatif arsitektur Big Data yang dapat digunakan selain Hadoop. Namun sejauh ini IPFS secara teori sudah mampu untuk mengatasi permasalahan Single Point of Failure (SPoF), hal ini dikarenakan IPFS adalah jaringan terdesentralisasi, dimana setiap peer yang mengunduh blok file menyimpan salinannya untuk sementara waktu, kemudian membagikannya dengan yang lain (Gingu *et al.*, 2022). Oleh karena itu, IPFS tidak memiliki satu titik pusat yang dapat mengalami kegagalan dan mengganggu keseluruhan sistem. Jika satu node mati atau tidak dapat diakses, data masih dapat diakses melalui node lain yang memiliki salinan dari data tersebut. Namun demikian ada aspek lain yang perlu dipertimbangkan yaitu performansi eksekusi data pada Blockchain dan IPFS.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka rumusan masalah pada penelitian ini adalah bagaimana menentukan performansi eksekusi data pada arsitektur Big Data dengan pemanfaatan Teknologi Blockchain dan Interplanetary File System (IPFS).



ujuan Penelitian

ujuan penelitian ini adalah merancang, menguji dan memperoleh nilai nsi eksekusi data pada Arsitektur Big Data dengan pemanfaatan i Blockchain dan Interplanetary File System (IPFS).

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan gambaran arsitektur Big Data dengan Teknologi Blockchain dan Interplanetary File System (IPFS) dan tersedianya sistem yang dapat menjadi alternatif yang dapat digunakan sebagai pengolahan Big Data.

1.5 Batasan Masalah

Batasan Masalah penelitian ini adalah sebagai berikut.

1. Penelitian ini berfokus pada evaluasi performansi eksekusi data yaitu Throughput, Latency dan waktu.
2. Blockchain yang digunakan adalah Blockchain Ethereum yang diintegrasikan dengan Interplanetary File System (IPFS)
3. Data yang digunakan sebagai uji coba adalah data file test yang dihasilkan secara online
4. Node yang digunakan sebanyak 5 (lima) node.

1.6 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang: latar belakang, rumusan masalah, tujuan penelitian, kegunaan penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini memuat tinjauan teori dan konsep dasar dari penelitian yang akan dilakukan berhubungan dengan *Big Data*, *Hadoop*, *Blockchain*, *Interplanetary File System (IPFS)*.



BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang tahapan penelitian, waktu dan lokasi penelitian, instrumen penelitian, tahap persiapan, gambaran umum sistem, struktur dataset, skenario pengujian, analisis waktu, dan analisis akurasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjabarkan proses penelitian yang telah dilakukan dan membahas hasil penelitian.

BAB VI KESIMPULAN DAN SARAN

Bab ini merupakan bab akhir yang berisi kesimpulan, implikasi, keterbatasan penelitian yang telah dilakukan, serta saran untuk penelitian selanjutnya.



BAB II TINJAUAN PUSTAKA

2.1 Big Data

Big data kumpulan data yang terus bertambah besar dan mencakup format yang heterogen, yang artinya Big data memiliki format data yang terstruktur, tidak terstruktur, dan semi terstruktur. Big data memiliki sifat yang kompleks yang membutuhkan teknologi canggih dan algoritma canggih (Oussous *et al.*, 2018).

Big data dan analisisnya merupakan pusat dari ilmu pengetahuan modern dan bisnis. Data ini dihasilkan dari transaksi online, email, video, audio, gambar, log, postingan. Data-data tersebut disimpan dalam basis data yang terus bertumbuh dan menjadi sulit untuk dikelola, menganalisa dan memvisualisasikan melalui perangkat lunak basis data tradisional (Sagiroglu *et al.*, 2013)

Big Data memiliki tiga komponen utama, yaitu *Variety*, *Velocity*, dan *Volume*.

Variety : Big data berasal dari berbagai macam sumber dan pada umumnya terdiri dari tiga jenis format data yaitu terstruktur, semi terstruktur, dan tidak terstruktur.

Volume : Jumlah data yang dihasilkan sangat besar dibandingkan dengan sumber data tradisional, ukuran data dapat lebih besar dari terabyte dan petabyte.

Velocity : Data dihasilkan dengan sangat cepat dengan proses yang tidak pernah berhenti dan diolah menjadi informasi yang bermanfaat

Selain ketiga komponen diatas, big data juga memiliki dua komponen tambahan. Kedua komponen tersebut adalah *Value* dan *Veracity*.

Value : Sebuah data dapat disebutkan memiliki value jika hasil dari pemrosesan data tersebut dapat membantu dalam pengambilan keputusan yang lebih baik.



: Mengacu pada tingkat akurasi dalam pengumpulan data dan seberapa akurat data tersebut. Dengan data yang memiliki tingkat keakuratan yang tinggi, pengambilan keputusan akan lebih baik dan maksimal (Setiawan., 2021)

Big data dapat menjadi referensi untuk mengembangkan sebuah produk. Informasi yang kira-kira dibutuhkan akan disimpan ke dalam big data dan hasil dari analisis tersebut dapat menjadi dasar untuk mengambil keputusan yang tepat untuk pengembangan bisnis. Selain pada sector bisnis big data juga bermanfaat pada bidang – bidang lainnya seperti:

Healthcare : Sistem pendukung keputusan klinis, analisis individual yang dapat diterapkan untuk profil pasien, obat, penetapan harga obat, dan menganalisis pola penyakit.

Sektor Publik : Pengambilan keputusan dengan sistem otomatis untuk mengurangi risiko dan berinovasi dalam produk dan layanan baru.

Manufaktur : Memperkirakan peningkatan permintaan , perencanaan rantai pasokan, dan dukungan penjualan (Sagioglu *et al.*, 2013)

2.2 Hadoop

Hadoop adalah kerangka kerja perangkat lunak open source berbasis Java dibawah lisensi Apache untuk komputasi besar secara intensif.

Hadoop adalah *framework software* berbasis java dan open source yang berfungsi untuk mengolah data yang sangat besar secara terdistribusi dan berjalan di atas cluster yang terdiri atas beberapa computer yang saling terhubung. Hadoop dapat mengolah data dalam jumlah yang sangat besar hingga petabyte dan dijalankan diatas ribuan computer. Hadoop mengatur segala macam proses detail sedangkan pengembang aplikasi hanya perlu focus pada logik aplikasinya. Hadoop bersifat open source dan berada dibawah Apache Software Fondation. Inti dari Hadoop terdiri atas:

- HDFS (*Hadoop Distributed File System*) yaitu sistem file yang terdistribusi.
- MapReduce yaitu *Framework* dari aplikasi yang terdistribusi

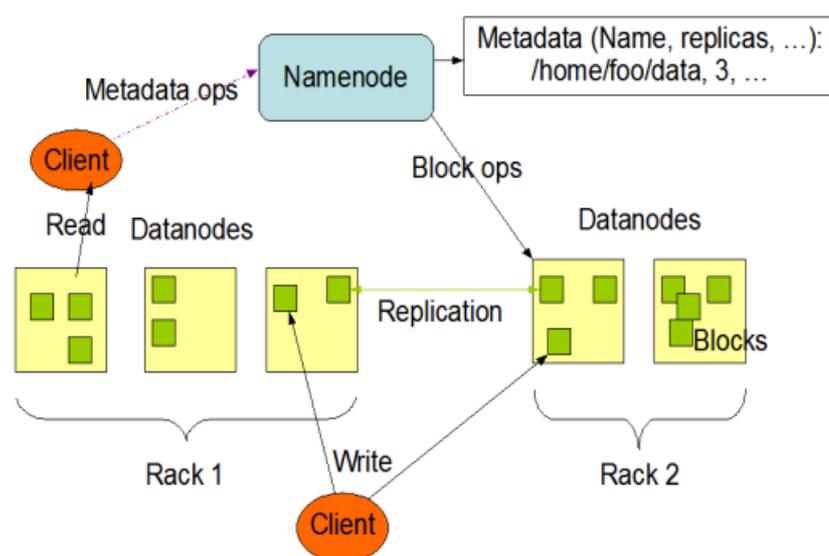
1. Hadoop Distributed File System (HDFS)

HDFS merupakan sistem penyimpanan terdistribusi, yang melakukan pemecahan file besar menjadi bagian-bagian lebih kecil kemudian



didistribusikan ke cluster-cluster dari komputer. HDFS terdiri dari beberapa jenis node di dalam cluster:

- Name Node : Node utama yang mengatur penempatan data di cluster, menerima job dan program untuk melakukan pengolahan dan analisis data misal melalui Map Reduce. Name Node juga menyimpan metadata tempat data di cluster dan juga replikasi data tersebut.
- Data Node : Node tempat data disimpan, satu block pada HDFS/data node adalah 64 MB.
- Secondary Name Node : Menyimpan informasi penyimpanan data dan pengolahan data yang ada di name node. Fungsinya jika Name Node mati dan diganti dengan Name Node baru maka Name Node baru bisa langsung bekerja dengan mengambil data dari Secondary Name Node.
- Checkpoint Node dan Backup Node : Checkpoint node melakukan pengecekan setiap interval waktu tertentu dan mengambil data dari Name Node. Dengan check point node maka semua operasi perubahan pada data terekam. Namun, secondary name node hanya perlu menyimpan check point terakhir. Backup Node juga berfungsi sama, hanya bedanya data perubahan yang disimpan dilakukan di memory bukan di file seperti checkpoint dan secondary node (Cholissodin *dkk.*, 2018)



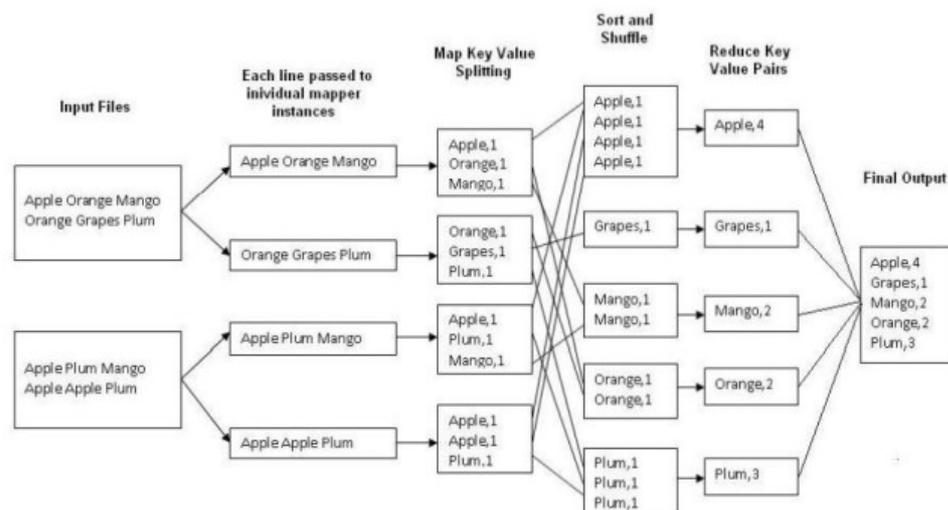
Gambar 2.1 Arsitektur HDFS



2.2.2. Map Reduce

MapReduce adalah sebuah model pemrograman yang didesain untuk dapat melakukan pemrosesan data dengan jumlah yang sangat besar dengan cara membagi pemrosesan tersebut ke beberapa tugas yang independen satu sama lain.

Model pemrograman Map Reduce membagi proses menjadi dua tahapan, yaitu tahapan Map dan tahapan Reduce. Map merupakan proses mengumpulkan semua informasi yang dimuat oleh potongan-potongan data dalam cluster di computer, dan kemudian akan disalurkan ke proses reduce. Dalam mengumpulkan data, Map akan membaca input key/value lalu menghasilkan output yang sama “key/value”, sedangkan Reduce merupakan proses membaca pasangan key/value yang dihasilkan Map, setelah membaca informasi key/value, Reduce akan menggabungkan setiap value dengan key yang sama menjadi satu kelompok. Selain itu, dua komponen yang terdapat pada Map Reduce yaitu, Job Tracker dan Task Tracker. Job Tracker adalah komponen pada komputer Master, dan Task Tracker adalah komponen pada komputer Slaves (Cholissodin *dkk.*, 2018).



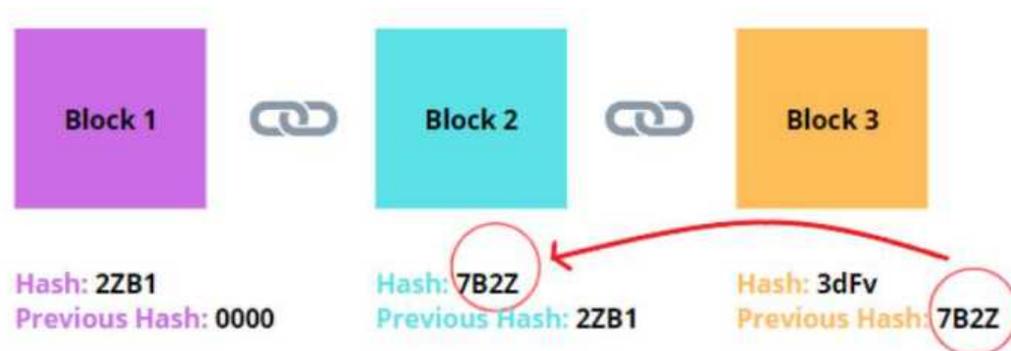
Gambar 2. 2 Alur MapReduce Untuk Kasus WordCount



2.3 Blockchain

Blockchain adalah buku besar terdistribusi yang tidak dapat diubah dan aman secara kriptografis dengan *fault tolerant* yang tinggi, sebagai teknologi yang mendasari sistem Bitcoin (Zheng *et al.*, 2018)

Blockchain dapat digambarkan sebagai penyimpanan data peer-to-peer yang sepenuhnya terdesentralisasi yang tersebar ke semua node. Blockchain didesain untuk menyimpan informasi yang tidak dapat diubah setelah data masuk kedalam rantai, dan oleh karena itu, blockchain merupakan sebuah basis data yang terdesentralisasi, terdistribusi, dan tidak dapat diubah dimana data terstruktur secara logis sebagai urutan potongan yang lebih kecil (blok). Setiap block $B_i > 0$ terhubung ke satu block sebelumnya B_{i-1} melalui fungsi hash kriptografi $H(B_{i-1})$. Perubahan pada B_{i-1} akan menghasilkan hash yang tidak valid pada B_i dan semua block berikutnya (Knirsch *et al.*, 2019).



Gambar 2. 3 Struktur Blockchain

Fitur utama blockchain dapat disimpulkan seperti berikut:

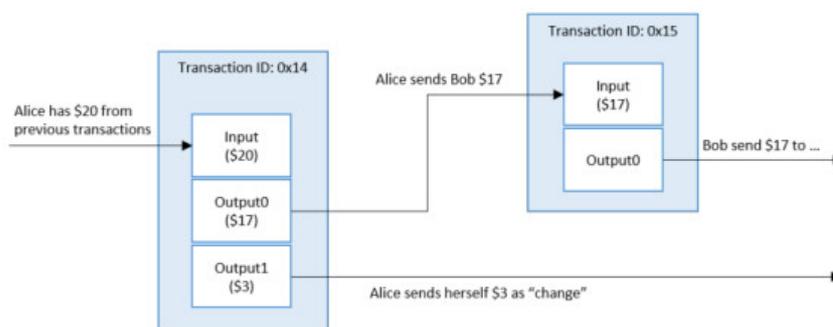
Decentralisasi : blockchain tidak hanya mengandalkan satu entitas/node terpercaya, melainkan beberapa atau semua entitas/node, hal ini bergantung pada algoritma consensus yang dipakai, ini berarti bahwa beberapa salinan data disimpan di semua node dan juga integritas data diatur oleh semua node.

Tidak dapat diubah : Setelah data disimpan ke dalam blockchain dan setiap node firmasi hal ini, data tersebut akan disimpan secara permanen dan tidak diubah. Mengubah data yang ada dalam block tertentu akan mengubah lock hingga block terakhir (Knirsch *et al.*, 2019).



Blockchain dapat dikategorikan berdasarkan model permissionnya, yang menentukan siapa yang dapat menghasilkan block baru. Jika semua orang dapat menghasilkan block baru, maka akan disebut *permissionless blockchain*. Jika hanya orang tertentu yang dapat menghasilkan block baru, maka akan disebut *permissioned blockchain*. Secara sederhana, *permissioned blockchain* seperti internet perusahaan yang dikontrol, sedangkan *permissionless blockchain* seperti internet public, dimana siapapun dapat berpartisipasi. *permissioned blockchain* pada umumnya digunakan oleh sekelompok organisasi atau individu yang disebut dengan konsorsium.

Dalam blockchain interaksi antar node disebut *transaction*, misalnya pada transaksi mata uang kripto, sebuah transaksi mewakili transfer mata uang kripto antar pengguna blockchain. Untuk scenario *business-to-business*, sebuah transaksi dapat menjadi sebuah cara untuk mencatat aktivitas yang terjadi pada asset digital atau fisik. Data yang terdapat dalam sebuah transaksi dapat berbeda untuk setiap implementasi blockchain, akan tetapi mekanisme transaksi pada umumnya sama. Informasi yang terdapat dalam transaksi dapat mencakup alamat pengirim, kunci publik pengirim, tanda tangan digital, input transaksi dan output transaksi (Yag *et al.*, 2018).



Gambar 2. 4 Transaksi pada Blockchain

Aspek penting dari blockchain adalah bagaimana menghasilkan block baru. dilakukan dengan mengimplementasikan salah satu dari sekian banyak consensus yang ada.



2.3.1. Proof of Work Consensus Model

Dalam model Proof of Work, seorang pengguna menghasilkan block baru dengan memecahkan teka teki komputasi, hal ini memungkinkan setiap node untuk memvalidasi block, setiap block yang tidak memenuhi teka teki akan ditolak (Yag *et al.*, 2018).

Proof of Work melibatkan pemindaian nilai yang Ketika di hash seperti hashing SHA-256, hash dimulai dengan sejumlah bit nol (Nakamoto., 2018). Sebagai contoh, dengan menggunakan hash SHA-256, node harus menemukan hash dengan nilai yang memenuhi target kriteria sebagai berikut:

$$\text{SHA256}(\text{"blockchain"}+\text{Nonce}) = \text{Hash Digest dimulai dengan "000000"}$$

Dalam contoh ini, string "blockchain" ditambahkan dengan nilai nonce dan kemudian hash digest dihitung. Nilai nonce hanya berupa nilai angka. Berikut beberapa output yang dihasilkan untuk menyelesaikan kasus tersebut:

$$\text{SHA256}(\text{"blockchain0"}) =$$

0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab9
38 (Tidak Terpecahkan)

$$\text{SHA256}(\text{"blockchain1"}) =$$

0xdb0b9c1cb5e9c680dff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10
(Tidak Terpecahkan)

.....

$$\text{SHA256}(\text{"blockchain10730895"}) =$$

0x000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d6758
7 (Terpecahkan)

Untuk menyelesaikan kasus ini, dibutuhkan 10.730.895 tebakan, dalam contoh kasus ini, setiap nilai "nol" tambahan akan meningkatkan kesulitan.

Setiap penambahan satu nilai "nol" didepan, node akan membutuhkan 224.175 tebakan pada perangkat yang sama.



Setelah node berhasil memecahkan teka teki Hashing, node akan mengirimkan block dengan nonce yang valid ke blockchain. Block akan diverifikasi bahwa block baru memenuhi persyaratan, kemudian menambahkan block baru ke Salinan blockchain dan mengirim ulang block ke node peer. Dengan cara ini, block baru didistribusikan dengan cepat ke seluruh jaringan node yang berpartisipasi (Yag *et al.*, 2018).

2.3.2. Proof of Stake Consensus Model

Berbeda dengan Proof of Work yang menggunakan komputasi untuk menghasilkan block baru, Proof of Stake menggunakan kripto atau uang digital yang dibekukan ke dalam blockchain, kegiatan ini disebut *Staking*. Proof of Stake menggunakan jumlah kripto yang dimiliki node sebagai faktor penentu dalam menghasilkan block baru. Dengan demikian, node blockchain menghasilkan block baru terkait dengan rasio kripto mereka terhadap keseluruhan jumlah jaringan blockchain dari kripto yang dibekukan.

Dengan model konsensus ini, tidak perlu melakukan lagi melakukan perhitungan intensif dengan melibatkan banyak sumber daya, karena konsensus ini menggunakan lebih sedikit sumber daya, sehingga mengatasi permasalahan pada konsensus Proof of Work (Saleh., 2018)

Metode bagaimana jaringan blockchain menggunakan *staking* dapat bervariasi, pada umumnya ada empat metode yang digunakan:

- Random Selection atau chain-based proof of stake : Jaringan blockchain akan melihat semua kripto yang dibekukan oleh user dan memilih berdasarkan rasio jumlah kripto user yang dibekukan, jadi jika user memiliki 42% dari seluruh jumlah kripto yang dibekukan pada jaringan blockchain, maka user akan di pilih 42% kali.
- Multi round voting system atau Byzantine fault tolerance proof of stake : jaringan blockchain akan memilih beberapa user untuk membuat block yang akan diusulkan, kemudian semua user akan memberikan voting untuk block yang diusulkan. Beberapa putaran voting dapat terjadi sebelum block baru diputuskan.



- Coin age system atau coin age proof of stake : kripto yang dibekukan memiliki properti age, setelah jangka waktu tertentu user kripto yang dibekukan dapat dipertimbangkan untuk menerbitkan block baru. Kemudian kripto yang dipilih property agennya akan diatur ulang dan tidak dapat digunakan lagi hingga beberapa waktu (Yag *et al.*, 2018).

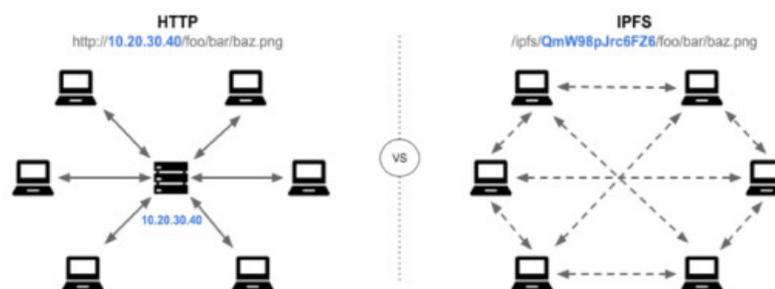
2.4 InterPlanetary File System (IPFS)

IPFS adalah protokol penyimpanan data terdistribusi. IPFS menghasilkan hash unik untuk setiap data yang disimpan (Zheng *et al.*, 2018). IPFS mengintegrasikan penggunaan struktur terkait Merkle yang kompleks dengan kemampuan pengalamatan data dari sistem berbagi file p2p. Data didistribusikan melalui jaringan p2p.

Protokol HTTP klasik menggunakan alamat lokasi, mengandalkan arsitektur terpusat di mana pengguna terhubung ke server pusat (lokasi) yang menyediakan file. Sebaliknya, IPFS menggunakan alamat konten, di mana pengguna dapat mengambil data yang diidentifikasi secara unik dari setiap node dalam jaringan terdistribusi yang menyimpan data tersebut (Suseno *dkk.*, 2022)

Menggunakan IPFS tidak hanya memungkinkan untuk mengakses file pada halaman web namun juga berbagai file yang disimpan pada komputer dalam bentuk dokumen, email, atau bahkan rekaman database. Sehingga dalam konsepnya terdapat tiga prinsip fundamental yang digunakan oleh IPFS :

- Identifikasi yang unik melalui content addressing
- Menghubungkan konten melalui directed acyclic graphs (DAGs)
- Menemukan konten melalui distributed hash tabels (DHTs)



Gambar 2. 5 Perbandingan HTTP dengan IPFS



IPFS dibagi menjadi beberapa subsystem dengan berbagai tujuan (Benet., 2014). Semua subsystem IPFS dapat dilihat pada tabel 2.1:

Tabel 2. 1 Tabel Subsytem IPFS

| Tujuan | Subsytem |
|---|--|
| Menghadirkan dan mengatur data | CIDs, IPLD, UnixFS, MFS, DAG-CBOR, DAG-JSON, CAR files |
| Content routing, menyambungkan antara CID dan alamat IP | Kademlia DHT, Delegated routing over HTTP, Bitswap, mDNS |
| Transfer data | Bitswap, HTTP Gateways, Sneakener, Graphsync |
| Pengalamatan data dan peers | Multiformats |
| Menjembatani antara IPFS dan HTTP | IPFS Gateways, Pinning API Spec |
| Konektivitas peer-to-peer | Libp2p (TCP, QUIC, WebRTC, WebTransport) |
| Mutabilitas dan Penamaan yang dinamis | IPNS (Interplanetary Naming System), DNSLink |

IPFS mempresentasikan data sebagai block beralamat konten, dan beroperasi pada block data tersebut menggunakan subsystem berikut:

Content Identifier (CID) : Dalam IPFS, data dipotong-potong menjadi beberapa blok, yang diberi pengenal unik yang disebut Content Identifier (CID). Secara umum, CID dihitung dengan menggabungkan hash data dengan codec-nya. Codec dibuat dengan menggunakan Multiformats.

InterPlanetary Linked Data (IPLD) : IPFS menggunakan InterPlanetary Linked Data (IPLD) untuk bekerja dengan CID dan data beralamat konten. IPFS menggunakan IPLD untuk merepresentasikan hubungan antara data beralamat konten, seperti direktori file dan struktur hirarkis lainnya, dengan menggunakan Directed Acyclic Graph (DAG) yang disebut Merkle DAG. Dengan menggunakan



tuk fungsionalitas umum, IPFS dapat menyediakan mekanisme yang sifik untuk merepresentasikan dan mengalamatkan berkas, direktori, dan nya, yang disebut UnixFS. Dengan UnixFS, IPFS dapat memecah dan

menautkan data yang terlalu besar untuk dimasukkan ke dalam satu blok, dan menggunakan representasi yang terpecah-pecah untuk menyimpan dan mengelola data.

Content Addressable aRchive (CAR) files : IPFS menggunakan file Content Addressable aRchive (CAR) untuk menyimpan dan mentransfer arsip berseri data beralamat konten IPLD. File CAR mirip dengan file TAR, yang dirancang untuk menyimpan koleksi data beralamat konten.

2.5 MetaMask

Metamask merupakan dompet cryptocurrency yang dapat digunakan di browser Chrome, Firefox dan Brave. Metamask juga merupakan ekstensi browser. Yang berarti bahwa Metamask bekerja seperti jembatan antara browser normal dan blockchain Ethereum.

Metamask memungkinkan pengguna untuk menyimpan dan mengelola kunci akun, menyiarkan transaksi, mengirim dan menerima token berbasis Ethereum dan terhubung dengan naman ke DApps melalui browser yang kompatibel atau browser bawaan aplikasi seluler.

Blockchain Ethereum merupakan jaringan tempat para pengguna dapat membangun aplikasi mereka sendiri yang disebut dApps dan cryptocurrency. 18 Ethereum juga memungkinkan penggunaanya untuk menulis pedoman transaksi yang disebut smart contract atau kontrak pintar.

Cryptocurrency atau mata uang kripto jaringan blockchain Ethereum disebut Ether. Mata uang kripto lain yang dibangun diatas jaringan blockchain Ethereum biasa disebut token. Sebagian besar token-token yang dibangun di jaringan blockchain Ethereum disebut dengan token ERC20 karena token-token yang dibangun mengikuti aturan yang telah ditetapkan oleh pengembang Ethereum untuk membuat cryptocurrency baru pada jaringan Ethereum (Suseno *dkk.*, 2022).



2.6 Smart Contract

Istilah “Smart Contract” diciptakan oleh Nick Szabo pada pertengahan tahun 1990-an. Dia menyarankan untuk menerjemahkan klausul “kontrak” ke dalam kode dan menanamkannya ke dalam perangkat lunak atau perangkat keras untuk membuatnya dapat dieksekusi sendiri, untuk meminimalkan biaya kontrak antara pihak-pihak yang bertransaksi dan untuk menghindari pengecualian yang tidak disengaja atau tindakan jahat selama pelaksanaan kontrak (Zheng *et al.*, 2019)

Smart Contract pada dasarnya disimpan, direplikasi, diperbaharui dan terdistribusi dalam blockchain. Seluruh siklus smart contract terdiri dari empat bagian, sebagai berikut:

Creation of Smart Contract : terdiri dari desain, implementasi dan validasi atau pengujian.

Deployment of Smart Contract : Smart Contract yang telah divalidasi kemudian di deploy ke dalam blockchain. Smart contract yang disimpan di dalam blockchain tidak dapat dimodifikasi. Setiap perubahan membutuhkan contract baru. Setelah smart contract dideploy pada blockchain, semua pihak dapat mengakses contract melalui blockchain.

Execution of Smart Contract : Setelah smart contract dideploy, Ketika kondisi contract tercapai pernyataan yang sesuai akan secara otomatis dieksekusi, akibatnya sebuah transaksi dieksekusi, dan divalidasi oleh node di blockchain.

Completion of Smart Contract : Setelah smart contract dieksekusi, status baru dari semua pihak yang terlibat akan diperbaharui, status yang diperbaharui akan disimpan dalam blockchain.

2.7 Apache JMeter

Apache JMeter adalah perangkat lunak open source, aplikasi Java murni 100% dirancang untuk memuat, menguji perilaku fungsional, dan mengukur



Awalnya dirancang untuk pengujian aplikasi web tetapi kemudian kearea pengujian lainnya. JMeter dapat digunakan di server, jaringan k untuk menguji kompresibilitas layanan dengan mensimulasikan beban

atau untuk menganalisa kinerja keseluruhan dibawah tekanan yang berbeda. JMeter dapat mendukung pengujian kinerja diberbagai bidang seperti web, database, FTP, LDAP, web service, JMS dll (Wang et al., 2019).

Pengujian JMeter pada umumnya melibatkan pembuatan loop dan group thread. Loop mensimulasikan permintaan berurutan ke server dengan penundaan yang telah ditentukan. Group thread dirancang untuk mensimulasikan beban bersamaan. Jmeter menyediakan antarmuka pengguna. JMeter juga menyediakan API yang memungkinkan untuk menjalankan pengujian berbasis JMeter dari aplikasi Java. Untuk membuat pengujian di JMeter, perlu dibuat rencana pengujian, yang pada dasarnya merupakan urutan operasi yang akan dijalankan Jmeter. Rencana pengujian yang paling sederhana mencakup elemen-elemen berikut :

1. Thread Group : elemen ini digunakan untuk menentukan jumlah thread yang sedang berjalan dan periode peningkatan. Setiap thread mensimulasikan seorang pengguna dan periode peningkatan rampu-up menentukan waktu untuk membuat semua thread.
2. Sampler : elemen ini merupakan permintaan yang dapat dikonfigurasi ke server HTTP, FTP, atau permintaan LDAP.
3. Listeners : elemen ini digunakan untuk mengirim data permintaan proses.

2.8 Tinjauan Empiris

Pada pembahasan ini, tinjauan empiris menguraikan beberapa penelitian terdahulu yang bersinggungan dengan penelitian ini. Sejumlah penelitian yang terkait dengan penelitian ini telah beberapa kali dilakukan, meskipun tidak secara spesifik meneliti variabel – variabel yang persis sama dengan penelitian ini. Selain itu, terdapat beberapa perbedaan yang terletak pada metode penelitian. Adapun penelitian – penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini sebagai berikut :



Tabel 2. 2 Tabel State Of the Art

| No. | Judul Karya Ilmiah, Nama, Tahun Terbit dan Penerbit | Objek dan Permasalahan | Metode Penyelesaian | Kinerja |
|-----|---|---|---|--|
| 1. | Judul : Analisa Teknologi Blockchain dan InterPlanetary File System (IPFS) untuk Pengoptimalan Arsitektur Big Data | Objek : Arsitektur big data, HDFS | Metode Penyelesaian : Menggunakan teknologi Blockchain, InterPlanetary File System (IPFS) | Kinerja : pemabahasan mengenai perbandingan eksekusi data berdasarkan waktu dan throughput pada masing masing arstiktur. |
| | Judul: Simplified HDFS Architecture with Blockchain Distribution of Metadata Penulis : Deepa S. nar, et.al Terbit : International | Objek: HDFS Permasalahan: Bagaimana membangun arsitektur untuk penyimpanan big data yang menghilangkan konsep master node | Blockchain | Kinerja : membuat blockchain sebagai media penyimpanan metadata, sehingga menghilangkan peran Namenode pada arsitektur big data, sehingga menghasilkan penundaan akses |



Journal of
Applied
Engineering
Research

Tahun :

2017

metadata yang rendah dan meningkatkan waktu eksekusi permintaan data pada Hadoop. Namun dilain sisi penggunaan blockchain untuk menyimpan metadata menimbulkan permasalahan pada skalabilitas disebabkan karena blockchain tidak mampu untuk menyimpan dan memproses data dalam jumlah besar

Judul:
Improving
HDFS Write
Performance
Using Efficient
Replica
Placement

Penulis :

Patel Neha M,

l

erbit :

Objek: HDFS

Permasalahan:
Bagaimana
meningkatkan
kinerja penulisan
HDFS

Menggunaka
n Replica
Placement
secara paralel

mereplikasi
setiap blok HDFS
pada node yang
berbeda,
sehingga
meningkatkan
throughput tulis
HDFS hingga
10%



 IEEE

Tahun :

2014

| | | | | |
|---|--|---|---|---|
| Judul: Ordinal Optimization-Based Performance Model Estimation Method for HDFS | Objek: HDFS Permasalahan: Bagaimana meningkatkan model kinerja HDFS yang optimal | Ordinal Optimization | Metode yang diusulkan efektif dan praktis untuk memperkirakan struktur terbaik dari model kinerja HDFS W/R berbasis Piecewise Linear. Ini tidak hanya menjamin solusi terbaik dengan probabilitas tinggi tetapi juga memberikan evaluasi yang konsisten secara global. | |
| 4. Tian Ma, et.al | Penerbit : | IEEE | Tahun : | |
| 2020 | Judul: Efficient Continuous Big Data Integrity Checking | Objek: Big Data Permasalahan: Bagaimana | Blockchain, Skema probabilistic CBDIC (Continuous | Mampu memeriksa beberapa file dalam selang waktu tertentu, |



| | | | |
|--|--|--|--|
| <p>5. Decentralized Storage</p> <p>Penulis : Haiyan yu, et.al</p> <p>Penerbit : IEEE</p> <p>Tahun : 2021</p> | <p>memeriksa integritas data besar secara berkelanjutan yang efisien untuk penyimpanan terdesentralisasi</p> | <p>Big Data Integrity Checking)</p> | <p>sehingga dapat memverifikasi integritas big data dengan probabilitas verifikasi yang tinggi dan mengatasi masalah efisiensi file yang harus diproses secara individual.</p> |
| <p>6. Judul: Decentralising Big Data Processing</p> <p>Penulis : Scott Ross Brisbane</p> <p>Penerbit : The University of New South Wales</p> <p>Tahun : 2016</p> | <p>Objek: Arsitektur Big Data</p> <p>Permasalahan: Bagaimana membangun arsitektur big data secara decentralisasi</p> | <p>Interplanetary File System (IPFS)</p> | <p>Berhasil memungkinkan analitik data berbasis lintas wilayah, peer-to-peer. Untuk kinerja local dan cross region HDFS lebih baik. Namun, ketika memperhitungkan waktu untuk mengambil kumpulan data, solusi IPFS jauh lebih cepat</p> |



| | | | | |
|--|--|---|-------------------------------|---|
| 7. | Judul: Optimizing Small File Storage Process of the HDFS Which Based on the Indexing Mechanism | Objek: HDFS Permasalahan: Bagaimana mengoptimalkan penyimpanan HDFS pada file dengan size yang kecil | Indexing | Skema indexing dapat meningkatkan efisiensi memori dan kecepatan membaca |
| Penulis : Wenjuan Cheng, et.al | | | | |
| Penerbit : IEEE | | | | |
| Tahun : 2017 | | | | |
| 8. | Judul: Secure Storage and Data Sharing Scheme Using Private Blockchain-Based HDFS Data Storage for Cloud Computing | Objek: HDFS Permasalahan: Data yang dikirimkan ke HDFS rentan terhadap serangan man-in-the-middle, dan beberapa serangan hacking, sehingga penting untuk meningkatkan keamanan pada | Blockchain Hyperledger Fabric | Hasil dari penelitian ini menunjukkan kinerja Algoritma AES lebih unggul daripada algoritma enkripsi lainnya yaitu berkisar 1,2 milidetik hingga 1,9 milidetik. Sebaliknya algoritma DES berkisar dari 1,3 |



| | | | |
|---|---|---|---|
| <p>Shrivasta, et.al Penerbit : International Journal of Computer Network and Application (IJCNA) Tahun : 2023</p> | <p>HDFS</p> | <p>milidetik hingga 3,6 milidetik, R C2 milidetik hingga 3,9 milidetik, dan RSA milidetik hingga 1,4 milidetik, dengan ukuran data mulai dari 910 kilo.</p> | |
| <p>9.</p> | <p>Judul: A Review on P2P File System Based on IPFS for Concurrency Control in Hadoop</p> <p>Penulis : Jasmine Sethi, et.al</p> <p>Penerbit : Springer</p> <p>Tahun : 2021</p> | <p>Objek: HDFS Interplanetary File System (IPFS)</p> <p>Permasalahan: Saat ini, jumlah data yang dihasilkan setiap menitnya sangat besar dan dapat diakses oleh banyak pengguna. Hal ini menciptakan masalah dalam mengelola, mengakses dan mengeksekusi data sehingga mengakibatkan masalah control konkurensi.</p> | <p>Hasil dari penelitian yang dilakukan yaitu mengimplementasikan antarmuka Filesystem berbasis Java Hadoop untuk sistem file terdesentralisasi peer-to-peer menggunakan IPFS. Antarmuka yang diusulkan akan memungkinkan fungsi Hadoop MapReduce untuk secara langsung dilakukan pada file data yang di-host di IPFS.</p> |



| | | | |
|--|---|------------------------------|---|
| <p>Judul: A Decentralised Framework for Efficient Storage and Processing of Big Data Usiing HDFS and IPFS</p> <p>Penulis : Franklin John, et.al</p> <p>Penerbit : International Journal of Humanitarian Technology</p> <p>Tahun : 2021</p> | <p>Objek: Big Data</p> <p>Permasalahan: Revolusi big data muncul dengan peluang dan tantangan yang lebih besar. Beberapa tantangan utama termasuk menangkap, menyimpan, mentransfer, menganalisis, memproses, dan memperbarui kumpulan data yang besar dan kompleks ini. Teknik penanganan data tradisional tidak dapat mengelola data yang berkembang pesat ini.</p> | <p>HDFS, dan IPFS</p> | <p>Sistem file terdistribusi Hadoop dan IPFS bersama-sama menangani tugas penempatan dan replikasi data dan kerangka kerja pemrograman MapReduce di Hadoop menangani tugas pemrosesan data. Hasil penelitian menunjukkan bahwa kerangka kerja yang diusulkan dapat mencapai penyimpanan yang hemat biaya serta pemrosesan data besar yang lebih cepat.</p> |
|--|---|------------------------------|---|

