

## ANALISIS PROSES SISTEM UNTUK IMPLEMENTASI *INTERPLANETARY FILE SYSTEM (IPFS)* PADA *SMART CONTRACT* ETHEREUM

### SYSTEM PROCESS ANALYSIS FOR IMPLEMENTATION OF *INTERPLANETARY FILE SYSTEM (IPFS)* IN *SMART CONTRACT ETHEREUM*

Jafar Haritsah<sup>1</sup>, Avon Budiyo<sup>2</sup>, Adityas Widjajarto<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom

<sup>1</sup>[jafarharitsah@student.telkomuniversity.ac.id](mailto:jafarharitsah@student.telkomuniversity.ac.id), <sup>2</sup>[avonbudi@telkomuniveristy.co.id](mailto:avonbudi@telkomuniveristy.co.id),

<sup>3</sup>[adtwjrt@telkomuniversity.ac.id](mailto:adtwjrt@telkomuniversity.ac.id)

#### Abstrak

*File sharing* memang sangat berguna untuk berkomunikasi dan juga bertukar data, data yang dimaksud bisa berupa angka, sumber, skala dan juga suatu informasi yang penting untuk dijaga keamanannya. Dikarenakan saat ini semakin banyak teknologi yang menunjang *file sharing*, maka salah satu solusi untuk menjaga kewanan dari suatu data adalah *Interpanetary File System (IPFS)*. *IPFS* hadir untuk memberikan solusi permasalahan sebagai sistem *file* terdistribusi *peer-to-peer* yang *secure* dengan menggunakan sistem *file* terenskripsi (*hash*) yang menghubungkan semua perangkat komputasi dengan sistem *file* yang sama.

*Blockchain* atau dapat disebut sebagai teknologi pembukuan terdistribusi *Distributed Ledger Technology (DLT)* merupakan suatu sistem untuk menyimpan informasi yang disebarluaskan pada komputer agar informasi lebih aman dan terjaga untuk mencegah terjadinya pengambilan data dari orang – orang yang ingin mengambil tanpa izin. Selain penyimpanannya yang *secure blockchain* juga berguna untuk menyimpan data yang lebih sedikit, seperti informasi nilai *hash*, transaksi dan juga *metadata*. Untuk mengimplementasikan *IPFS* pada *smart contract* ethereum kita harus menganalisis proses sistem nya untuk mengetahui bagaimana proses transaksi maupun penyimpanan data, *smart contract* yaitu sistem untuk membantu dalam menerima dan mengirimkan data. *Ethereum* yaitu sebuah *platform* berbasis *blockchain*. Oleh karena itu, penelitian ini dilakukan untuk mengetahui proses sistem untuk *IPFS* agar *smart contract* ethereum dapat berjalan dengan baik dan sebagaimana mestinya.

**Kata Kunci :** *Blockchain, Smart Contract, Ethereum, Peer-to-peer, Proses Sistem*

#### Abstract

*File sharing* is really very useful for communicating and also exchanging data, the data in question can be in the form of numbers, sources, scales and also information that is important to maintain security. Because currently more and more technology is supporting file sharing, one of the solutions to safeguarding data from is *Interpanetary File System (IPFS)*. *IPFS* is here to provide solutions to problems as a secure distributed *peer-to-peer* file system using an encrypted file system (*hash*) that connects all computing devices with the same file system.

*Blockchain* or can be referred to as bookkeeping technology distributed *Distributed Ledger Technology (DLT)* is a system for storing information that is disseminated on a computer so that information is safer and safer to prevent data retrieval from people who want to take it without permission. Besides the storage, the *secure blockchain* is also useful for storing less data, such as *hash* value information, transactions and also *metadata*. To implement *IPFS* on *smart contract* ethereum we have to analyze the system process to find out how the transaction process and data storage, *smart contract*, is a system to assist in receiving and sending data. *Ethereum* is a *blockchain* based platform. Therefore, this study was conducted to determine the system process for *IPFS* so that the *smart contract* ethereum can run properly and properly.

**Key Word :** *Blockchain, Smart Contract, Ethereum, Peer-to-peer, Process System*

## 1. Pendahuluan

---

Untuk melakukan suatu pertukaran data atau informasi dibutuhkan komunikasi antar satu komputer dengan komputer lainnya yang saat ini sudah semakin canggih yaitu dengan menggunakan *file sharing*. *File Sharing* merupakan sebuah penyediaan dan juga penerimaan *file digital* melalui sebuah jaringan yang menggunakan model terpusat atau sering disebut *peer to peer* (P2P), *file* disimpan dan dikelola oleh *personal* komputer pada suatu *user*. Selain untuk berbagi *file*, *file sharing* juga bisa digunakan untuk penggunaan internet dalam berbagi *file* dengan pengguna internet lainnya. Semua kegiatan bisnis dilakukan melalui pembuatan kontrak. *Smart contract* Ethereum menjadi solusi permasalahan kontrak tertulis. *Smart contract* merupakan protokol komputer yang dimaksudkan untuk memfasilitasi, memverifikasi, atau menegakkan negosiasi atau kinerja suatu kontrak secara digital melalui kode program. Kontrak pintar memungkinkan kinerja transaksi yang *kredibel* tanpa pihak ketiga tanpa bisa dilacak dan diubah. Pemanfaatan *smart contract* misalnya dapat digunakan untuk produk dan layanan finansial, royalti konten, melacak kepemilikan berbagai jenis properti, verifikasi identitas *digital* ataupun pemungutan suara, dan lain lain [1].

*Blockchain* merupakan sistem basis data yang terus berkembang yang diamankan menggunakan teknik kriptografi, *blockchain* juga disebut sebagai sebuah buku besar yang terdistribusi untuk mencatat transaksi antara dua pihak secara efisien dan dengan cara yang dapat diverifikasi dan permanen, untuk mencatat transaksi digital yang terpusat dibutuhkan sistem *Distributed Ledger Technology* (DLT), yang digunakan untuk membuat *smart contract* atau *blockchain contract* [2]. Akan tetapi menyimpan data pada *Blockchain* memerlukan biaya yang cukup mahal. Karena *Blockchain* tidak cocok untuk menyimpan data dalam jumlah besar maka developer membuat sistem DApp yang mengkombinasikan IPFS pada *smart contract* Ethereum.

Pada penelitian ini akan dibangun sistem aplikasi DApp berbasis web yang mengkombinasikan teknologi *Smart Contract* Ethereum dan IPFS. Untuk menjalankan DApp yang telah dibuat pada sistem operasi membutuhkan proses sistem dan sumber daya CPU agar DApp dapat beroperasi sebagai mana mestinya. Penelitian yang dilakukan untuk mengukur proses sistem dan sumber daya CPU yang digunakan DApp. Pada penelitian Tugas Akhir ini penulis menggunakan metode *Network Development Life Cycle* (NDLC) yaitu suatu metode yang berguna untuk mengembangkan dan merancang jaringan infrastruktur yang terdiri dari *analysis, design, simulation prototyping, implementation, monitoring, dan management* untuk melakukan pemantauan jaringan agar mengetahui bagaimana kinerja jaringan dan juga statistic pada jaringan tersebut [3].

## 2. Dasar Teori

### 2.1. Peer-to-Peer

*Peer to peer* (P2P) merupakan sebuah paradigma atau pemodelan jaringan di mana setiap *peer* (sebutan untuk setiap komputer) yang mampu terhubung dan ikut serta berkontribusi di dalam penyediaan layanan dan pertukaran data.

### 2.2. Blockchain

*Blockchain* adalah sebuah teknologi pencatatan transaksi yang terintegrasi menggunakan teknologi *modern* yang dimana memiliki kode unik yang kekal (tidak bisa diubah) dalam merevolusi cara kerja internet, perbankan dan lain-lain.

### 2.3. InterPlanetary File System (IPFS)

IPFS adalah sistem *file* terdistribusi *peer to peer* yang menghubungkan semua perangkat komputasi dengan sistem *file* yang sama. Pengiriman konten terdistribusi akan menghemat *bandwidth* dan mencegah serangan DDoS [4].

### 2.4. Proses Sistem

Proses adalah program yang sedang dieksekusi/*running*. Proses merupakan unit kerja terkecil dalam sistem operasi. Proses melewati serangkaian *state*. Dan beragam kejadian dapat terjadi yang menyebabkan perubahan *state* pada proses. [5].

### 2.5. Network Development Life Cycle (NDLC)

*Network Development Life Cycle* yaitu suatu metode yang digunakan dalam mengembangkan dan merancang jaringan infrastruktur yang dapat digunakan sebagai observasi pada jaringan sehingga dapat mengetahui *statistic* dan kinerja pada jaringan.

### 2.6. Htop

Htop adalah sistem untuk memonitor proses pada program yang berjalan secara interaktif, htop memberikan informasi visual tentang prosesor, CPU dan memori yang sedang berjalan. Htop dirancang sebagai alternatif program Unix yang memberikan daftar lengkap proses yang berjalan [6].

### 2.7. Metamask

Metamask adalah salah satu aplikasi desentralisasi yang tertua dan banyak digunakan pada jaringan Ethereum. MetaMask merupakan ekstension dari chrome, berfungsi sebagai tempat penyimpanan Ether.

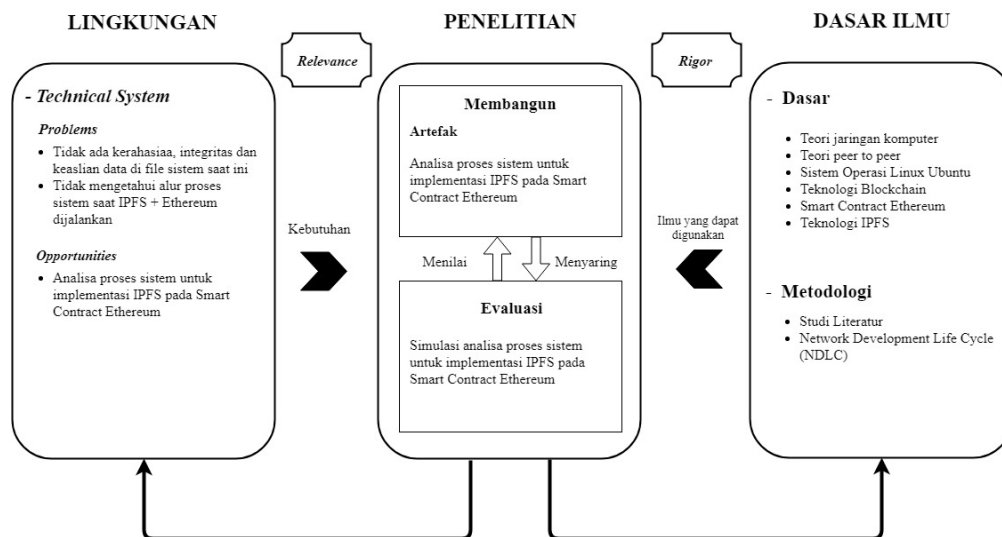
2.8 Node Package Manager (NPM)

NPM adalah Salah satu tool yang akan sering digunakan dalam Nodejs dalam mendvelop JavaScript. Terdapat beberapa package yang dibutuhkan untuk membantu penelitian ini, yaitu create-react-app, react-bootstrap, fs-extra, ipfs-api dan web3 [7].

3. Metode Penelitian

3.1. Model Konseptual

Model Konseptual adalah suatu cara kerja dalam melakukan langkah-langkah kerja dalam penelitian. Model konseptual adalah suatu kerangka kerja yang menerangkan tentang serangkaian ide global untuk membedakan antara yang penting dan tidak penting. Model konseptual yang ada dalam penelitian ini adalah sebagai berikut:



Gambar 1 Model Konseptual

4. Perancangan Sistem dan Skenario Pengujian

Perancangan sistem yang dibuat untuk melakukan simulasi dibutuhkan instrumen pengujian untuk mendukung jalannya penelitian. Ada dua Instrumen yang dibutuhkan pada penelitian ini, yaitu instrumen fisik dan instrumen program yang keduanya akan diperinci sebagai berikut :

4.1 Instrumen Fisik

Instrumen fisik merupakan suatu perangkat keras yang digunakan untuk mendukung berjalannya simulasi pengujian. Berikut daftar instrumen fisik yang digunakan dan diperinci pada Tabel

Tabel 1 Instrumen Fisik

No.	Komponen	Spesifikasi	
1.	Laptop Asus VivoBook 14 A442U	Prosesor	Intel® Core™ i5- 8250U CPU @ 1.60 GHz 1.80 GHz
		Memori	8GB DDR4 2133MHz SDRAM
		Storage	1TB SATA HDD 5400RPM
		Grafis	Discrete graphics Nvidia GT 930MX 2GB / Nvidia GT 940MX 2GB
		Resolusi	14" HD (1366×768) / FHD (1920×1080)
2.	Laptop Asuspro essential PU451LD	Prosesor	Intel® Core™ i5 4210U Processor
		Memori	DDR3L 1600 MHz SDRAM (8192 MB RAM)

		<i>Storage</i>	1TB HDD 5400 RPM
		<b>Grafis</b>	NVIDIA® GeForce® 820M (2048 MB)
		<b>Resolusi</b>	14.0" 16:9 HD (1366x768)
3.	Laptop Lenovo B490	<b>Prosesor</b>	Intel® Core™ i3-3110M Processor (2.4 GHz, Cache 3MB)
		<b>Memori</b>	2 GB DDR3
		<i>Storage</i>	500 GB Serial ATA 5400 RPM
		<b>Grafis</b>	NVIDIA Geforce 705M 1GB
		<b>Resolusi</b>	14" WXGA LED
4.	MikroTik Routes RB751G- 2HND	<i>Architecture</i>	MIPS-BE
		<b>CPU</b>	AR7241 400MHz
		<i>Main Storage/NAND</i>	64MB
		<b>RAM</b>	64MB
		<i>LAN Ports</i>	5
		<b>Gigabit</b>	Yes
		<i>Switch Chip</i>	1
		<i>Integrated Wireless</i>	1
		<b>Wireless Standarts</b>	802.11 b/g/n
		<b>Wireless Tx Power</b>	30dbm
		<b>Integrated Antenna</b>	Yes
		<b>Antenna Gain</b>	2 x 2,5dBi

#### 4.2 Instruman Program

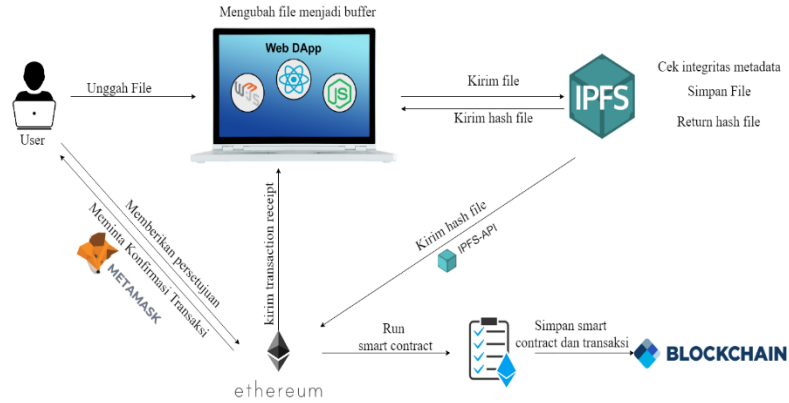
Instrumen program merupakan perangkat lunak yang mendukung jalannya simulasi pengujian. Berikut daftar dari instrumen program yang digunakan dan diperinci pada table sebagai berikut :

Tabel 2 Instrumen Program

<b>Type</b>	<b>Nama Software</b>	<b>Versi</b>
<b>Sistem Operasi</b>	<b>Linux Ubuntu</b>	<b>16.04 LTS</b>
<b>Main Software</b>	<b>IPFS (Interplanetary File System)</b>	<b>go-ipfs v0.4.18</b>
	<b>Go-lang</b>	<b>go1.12</b>
	<b>Node Package Manager (NPM)</b>	<b>6.4.1</b>
	<b>Node JS</b>	<b>V10.15.3</b>
	<b>Create-react-app</b>	<b>3.0.0</b>
<b>Dependency NPM</b>	<b>bootstrap</b>	<b>3.3.7</b>
	<b>fs-extra</b>	<b>7.0.1</b>
	<b>Ipfs-api</b>	<b>26.1.2</b>
	<b>React</b>	<b>16.8.6</b>
	<b>React-bootstrap</b>	<b>0.32.4</b>
	<b>React-dom</b>	<b>16.8.6</b>
	<b>react-script</b>	<b>3.0.0</b>
	<b>Web3</b>	<b>1.0.0-beta.34</b>
<b>Third party Software</b>	<b>Wireshark</b>	<b>2.6.3</b>
	<b>Google Chrome</b>	<b>74.0.3729.131 (64-bit)</b>
	<b>Metamask</b>	<b>6.4.1</b>
	<b>Sublime Text</b>	<b>3.2.1, Build 3207</b>

4.3 Arsitektur Sistem

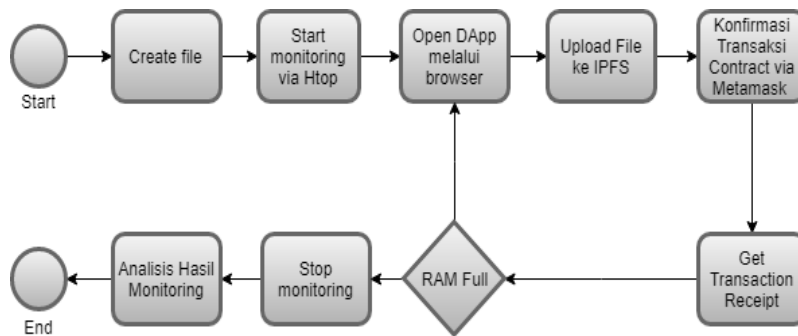
Pada penelitian ini dibangun kemudian dikembangkan suatu sistem yang berbasis web, yaitu *Decentralized Application* yang merupakan suatu penggabungan antara teknologi *Smart Contract* Ethereum dan *Interplanetary File System* (IPFS).



Gambar 2 Arsitektur Sistem

4.4 Skenario Pengujian

Pengujian dilakukan dengan cara memonitor proses sistem yang ada pada aplikasi Htop, yaitu *Task*, *Thread*, *Aplikasi Run*, *Load Average* dan *Average Time*. Hasil pengujian *Average Time* didapatkan dari penelitian lain mengenai komunikasi antar node IPFS pada *Smart Contract* Ethereum [14]. Selain membahas tentang proses sistem yang ada pada aplikasi Htop, pengujian ini juga membahas penggunaan CPU yang berjalan pada saat transaksi dilakukan, ada enam aplikasi yang akan dimonitoring menggunakan Htop yaitu, IPFS Daemon, Metamask, Aplikasi sistem web DApp, NPM, react script dan start.js.



Gambar 3 Skenario Pengujian

1. Skenario Pengujian Proses Sistem dan CPU Usage I  
Pada skenario I, dilakukan pengiriman transaksi *file* dari 1 node dan 2 node menerima.
2. Skenario Pengujian Proses Sistem dan CPU Usage II  
Pada skenario II, dilakukan pengiriman transaksi *file* dari 2 node dan 3 node menerima.
3. Skenario Pengujian Proses Sistem dan CPU Usage II  
Pada skenario II, dilakukan pengiriman transaksi *file* dari 2 node dan 3 node menerima.

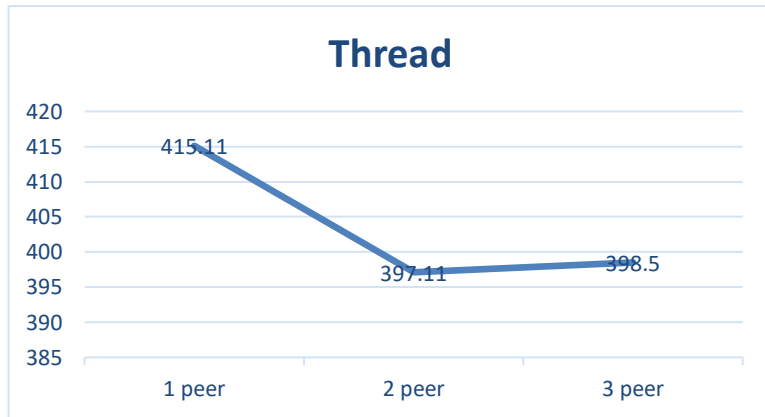
### 5. Pengujian dan Analisis Sistem

#### 1. Hasil pengujian Proses Sistem



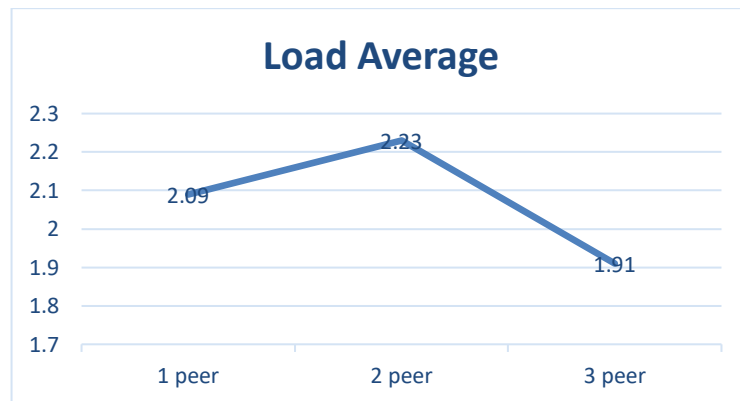
Gambar 4 Grafik Rata-Rata Proses *Task*

Gambar 4 adalah grafik rata-rata proses *task* yang dianalisis pada Htop. Pada analisis 1 peer mendapatkan hasil paling tinggi dengan angka 157.72, 2 peer 149.77 dan 3 peer 146.18.



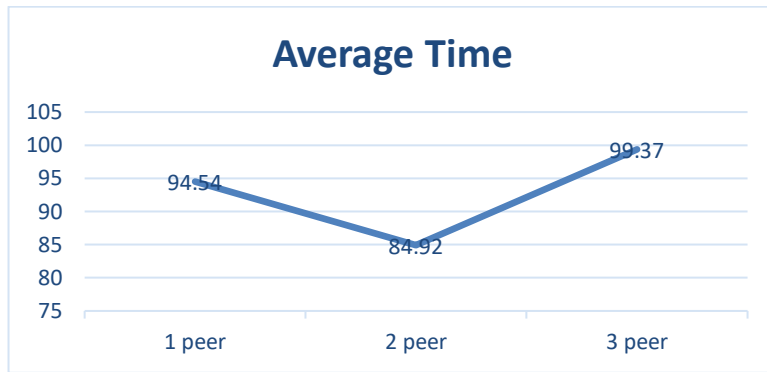
Gambar 5 Grafik Rata-Rata Proses *Thread*

Gambar 5 adalah grafik rata-rata proses *thread* yang dianalisis pada Htop. Pada analisis 1 peer mendapatkan hasil paling tinggi dengan angka 415.11, 2 peer 397.11 dan 3 peer 398.5.



Gambar 6 Grafik Rata-Rata Proses Load Average

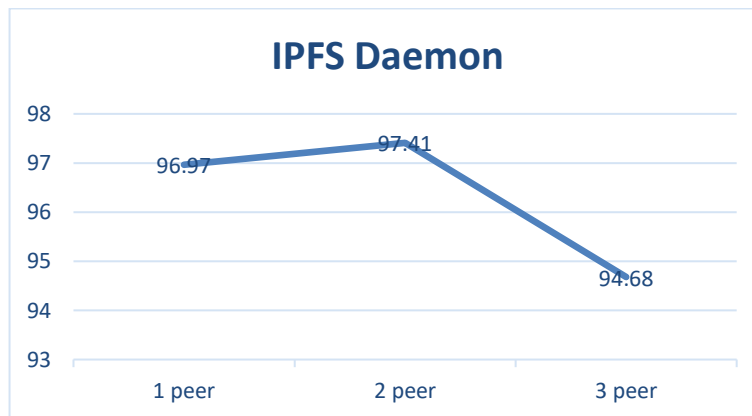
Gambar 6 adalah grafik rata-rata proses *load average* yang dianalisis pada Htop. Pada analisis 1 peer mendapatkan hasil dengan angka 2.09, 2 peer 2.23 dan 3 peer 1.91.



Gambar 7 Grafik Rata-Rata Proses Average Time

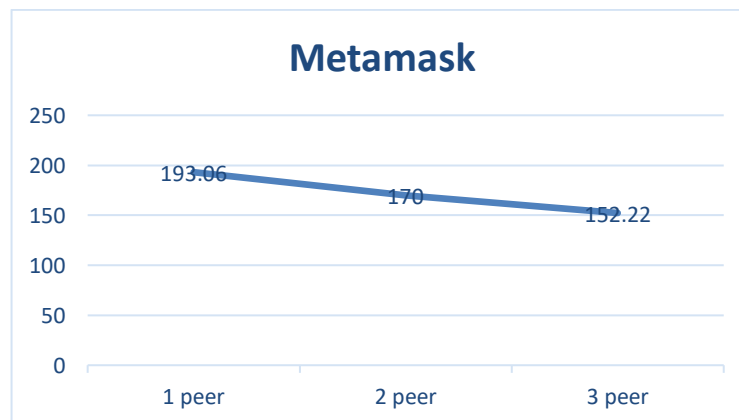
Gambar 6 adalah grafik rata-rata proses *load average* yang dianalisis pada Htop. Pada analisis 1 peer mendapatkan hasil dengan angka 94.54, 2 peer 84.52 dan 3 peer 99.37.

## 2. Hasil Pengujian CPU Usage



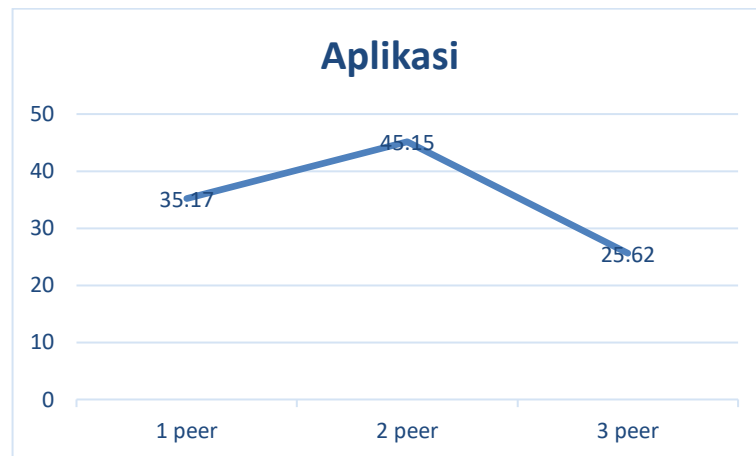
Gambar 8 Grafik Rata-Rata IPFS Daemon

Gambar 8 adalah grafik rata-rata proses pada aplikasi IPFS daemon yang dianalisis menggunakan Htop. Pada analisis 1 peer mendapatkan hasil dengan angka 96.97, 2 peer 97.41 dan 3 peer 94.68.



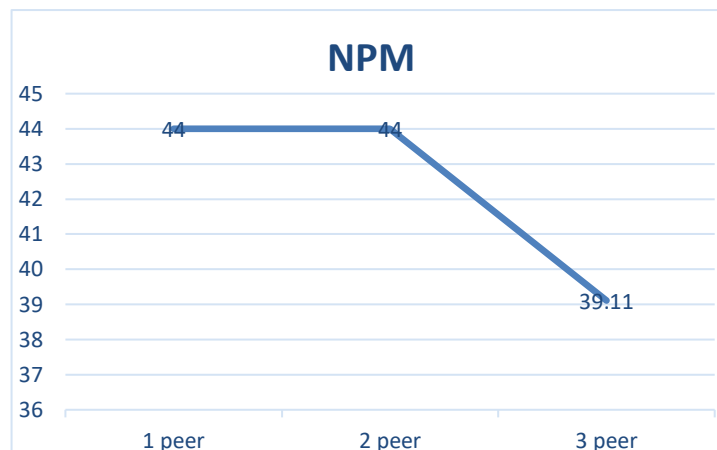
Gambar 9 Grafik Rata-Rata Metamask

Gambar 9 adalah grafik rata-rata proses pada aplikasi metamask yang dianalisis menggunakan Htop. Pada analisis 1 peer mendapatkan hasil tertinggi dengan angka 193.06, 2 peer 170 dan 3 peer 152.22.



Gambar 10 Grafik Rata-Rata Aplikasi Sistem Web DApp

Gambar 10 adalah grafik rata-rata proses pada aplikasi Sistem Web DApp yang dianalisis menggunakan Htop. Pada analisis 1 peer mendapatkan hasil dengan angka 35.17, 2 peer 45.15 dan 3 peer 25.62.



Gambar 10 Grafik Rata-Rata Aplikasi NPM

Gambar 10 adalah grafik rata-rata proses pada aplikasi NPM yang dianalisis menggunakan Htop. Pada analisis 1 peer mendapatkan hasil dengan angka 44, 2 peer 44 dan 3 peer 39.11.

## 6. Kesimpulan dan Saran

### 1. Kesimpulan

Spesifikasi pada setiap laptop mempengaruhi performa pengujian pada saat melakukan transaksi *file*, sehingga terjadinya perbedaan kualitas pada setiap perangkat, ada enam proses yang dimonitor pada CPU *usage* yang ada pada Htop, yaitu IPFS daemon, metamask, aplikasi sistem web DApp, NPM, react script dan start.js. Terdapat empat proses yang berjalan dengan baik dikarenakan empat proses tersebut memakai CPU untuk melakukan transaksi file, dan terdapat dua proses yang tidak berjalan dikarenakan dua proses tersebut hanya memakai memori pada saat melakukan transaksi file.

### 2. Saran

Pada pengimplementasian IPFS spesifikasi setiap node seharusnya setiap node memiliki spesifikasi yang sama, agar *resource* yang digunakan terbagi dengan sama rata. Pada proses sistem yang dianalisis seharusnya ada sistem pendukung untuk memonitoring proses tersebut agar lebih detail dalam melakukan pengujian.



## 7. Daftar Referensi

- [1] A. Kadir, *Konsep dan Tuntunan Praktis Basis Data*. Yogyakarta: AndiYogyakarta, 2002.
- [2] Amritha, Sindhu, *A Blockchain and IPFS based framework for secure Research record keeping*, 2012.
- [3] Kosasi, S Penerapan *Network Development Life Cycle* untuk Pengembangan Teknologi *Thin Client* pada Pendidikan KSM Pontianak. *Jurnal JIKE*, Vol. 1(2), 128, 2013.
- [4] Wennergen, O., dkk, A Blockchain-Based Framework for Data Sharing with Fine-grained Access Control in Decentralized Storage Systems. *Jurnal IEEE Access*, 4, 1-5. doi:10.1109/access.2018.2851611, 2018.
- [5] Setiawan, R, *Sistem Operasi*. Seribu Bintang, 2017.
- [6] Ramdani, Hairul, Implementasi dan analisis performa jaringan diskless system standar dengan diskless system cluster, 2013.
- [7] Dian, Muhar, Mengenal NPM untuk Manajemen Project Javascript, 2017.