JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 9
No. 2
Agustus
2023

Implementasi IPFS untuk Mengurangi *Gas Fee Smart Contract* Ethereum pada Aplikasi Penggalangan Dana

Hutomo Sakti Kartiko[#1], Tedy Rismawan[#2], Ikhwan Ruslianto[#3]

[#]*Program Studi Rekayasa Sistem Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura*
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak, 78124

[1]hutomosakti@student.untan.ac.id

[2]tedyrismawan@siskom.untan.ac.id

[3]ikhwanruslianto@siskom.untan.ac.id

***Abstrak*— *Blockchain* merupakan teknologi buku besar yang bersifat *decentralized*. *Blockchain* memiliki protokol *consensus* sebagai kesepakatan bersama dalam pengelolaan basis data. Contoh penerapan *blockchain* yaitu ethereum. Kelebihan ethereum yaitu dapat menjalankan program atau aturan yang disebut sebagai *smart contract*. Proses perubahan data pada ethereum memerlukan biaya transaksi atau *gas fee*. Nilai *gas fee* ini fluktuatif menyesuaikan *gas fee* terendah saat ini, kepadatan jaringan dan kompleksitas transaksi. *Smart contract* ethereum tidak efisien untuk menyimpan data yang berukuran besar karena semakin besar data yang disimpan maka semakin kompleks transaksi yang perlu dilakukan. Untuk meningkatkan efisiensi *gas fee smart contract* maka dilakukan sebuah penelitian dengan menerapkan *InterPlanetary File System* (IPFS). Teknik yang digunakan yaitu mengkombinasikan teknologi IPFS dengan *smart contract* ethereum untuk mengurangi kompleksitas transaksi ketika proses penyimpanan data penggalangan dana ke *smart contract* ethereum. Penerapan IPFS pada aplikasi penggalangan dana membutuhkan *gas fee* 0,00311847-0,003379868 ETH dengan kecepatan transaksi 12-36 detik. Berdasarkan pengujian sebanyak 40 kali dengan data yang berbeda, penerapan IPFS dapat menurunkan *gas fee* dengan rata-rata hingga 94,39% dan kecepatan transaksi sistem yang menerapkan IPFS lebih besar 13,55% dari sistem yang tidak menerapkan IPFS.**

***Kata kunci*— Ethereum, *Smart Contract*, Penggalangan Dana, IPFS.**

I. PENDAHULUAN

Teknologi *Blockchain* diperkenalkan secara publik oleh Nakamoto pada tahun 2008 dalam konteks *Bitcoin*, mata uang digital yang terdistribusi [1]. *Blockchain* diciptakan untuk merombak skema sirkulasi perantara dalam melakukan transaksi [2]. Transaksi antara A dan B bisa terjadi tanpa perantara, waktu yang lebih singkat, biaya yang lebih murah, dan bahkan jauh lebih aman dibandingkan transaksi dengan keterlibatan pihak ketiga yang sering menimbulkan permasalahan [3].

Blockchain merupakan jenis teknologi ledger terdistribusi yang paling umum yang ada saat ini, di mana dasar dari setiap transaksi yang ada tersebar di antara berbagai node yang dihubungkan oleh jaringan peer-to-peer. Ada beberapa perbedaan antara teknologi blockchain dan teknologi ledger terdistribusi konvensional. Perbedaan ini terlihat pada struktur data yang digunakan untuk pencatatan transaksi; di blockchain, setiap data transaksi dikirim ke blok yang hanya ditambahkan dan selalu terhubung, yang mencegah perubahan blok [4].

Salah satu contoh penerapan *blockchain* yaitu pada *Smart Contract*. *Smart contract* merupakan wadah yang menyimpan banyak *value* yang dapat terbuka jika suatu kondisi terpenuhi [5]. Ethereum adalah salah satu platform *blockchain* yang bisa digunakan untuk membuat *smart contract*. Ethereum merupakan jaringan *blockchain* yang digunakan untuk membuat *smart contract* menggunakan bahasa pemrograman *Turing-Complete* [6]. Ethereum memungkinkan para pengembang untuk membuat *Decentralized Application* (DApp) dengan hanya menulis kondisi dalam beberapa baris kode, sehingga aplikasi dapat berjalan sesuai dengan kondisi yang telah dibuat [7].

Smart contract pada jaringan *ethereum* tidak efisien untuk menyimpan data yang berukuran besar karena semakin besar data yang disimpan maka semakin besar juga biaya yang digunakan, sehingga perlu dikombinasikan bersama dengan teknologi *InterPlanetary File System* (IPFS) [8]. IPFS merupakan *Decentralized Cloud Storage* berbasis *blockchain* yang digunakan untuk menyimpan data digital yang berukuran besar dan juga dapat digunakan untuk melacak jejak digital data dari versi orisinil hingga versi terbaru [9].

Salah satu penelitian tentang penerapan *smart contract* pada sistem *crowdfunding* yaitu "*Blockchain Based Crowdfunding Systems in Malaysian Perspective*" [10]. Penelitian ini bertujuan untuk memecahkan beberapa masalah yang sering terjadi pada platform *crowdfunding*, seperti *campaign* yang tidak diatur dan beberapa *campaign*

ternyata penipuan. Selain itu, penyelesaian beberapa proyek *crowdfunding* juga mengalami masalah keterlambatan waktu. Hasil dari penelitian ini yaitu berhasil membuat sistem yang menerapkan *smart contract* pada jaringan *ethereum* sehingga transaksi berjalan dengan transparan dan *contract* yang telah dibuat akan sepenuhnya berjalan otomatis ketika semua persyaratan kontrak terpenuhi.

Selain itu, penelitian tentang pengurangan *gas smart contract* sudah pernah dilakukan sebelumnya dengan judul "*Quality of Service Ethereum Blockchain* berbasis IPFS untuk Validasi Ijazah Sekolah" [8]. Penelitian ini bertujuan untuk mengetahui *quality of service* dan *cost* dari *smart contract* ethereum. Dengan menerapkan *smart contract* berbasis IPFS dapat membuat *cost smart contract* menjadi lebih baik dibandingkan dengan tidak menggunakan IPFS. Hasil dari penelitian ini yaitu *smart contract* yang menggunakan IPFS lebih murah sebesar 61,95% daripada yang tidak menerapkan IPFS dan dengan biaya *gas* yang stabil.

Penelitian lain yang menggunakan IPFS yaitu "*Bountychain: Toward Decentralizing a Bug Bounty Program with Blockchain and IPFS*" [11]. Hasil dari penelitian ini yaitu menunjukan bahwa dengan mengimplementasikan IPFS dan *smart contract* dapat menghemat sekitar 21.164 *gas* dengan pengurangan sebesar 19,6% dari yang tidak mengimplementasikan IPFS. Berdasarkan penelitian terdahulu maka dilakukanlah penelitian dengan mengkombinasikan teknologi IPFS dan *smart contract* ethereum untuk membuat sistem penggalangan dana.

Penelitian ini bertujuan untuk mengetahui tingkat pengurangan *gas smart contract* ethereum saat menyimpan data pada sistem penggalangan dana yang menerapkan IPFS. Penerapan IPFS dapat menjadi solusi untuk mengurangi *gas fee* yang diperlukan ketika menyimpan data pada *smart contract* ethereum. Sehingga dilakukan penelitian dengan judul "Implementasi IPFS Untuk Mengurangi *Gas Fee Smart Contract* Ethereum Pada Aplikasi Penggalangan Dana". Dengan adanya penelitian ini diharapkan dapat membantu para pengembang yang akan menggunakan *smart contract* ethereum untuk mengurangi *gas fee smart contract*.

II. LANDASAN TEORI

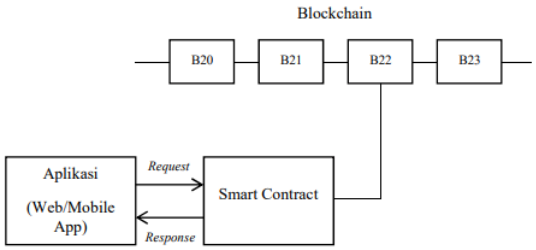
A. *Blockchain*

Teknologi Blockchain pada awalnya dikembangkan dalam bentuk Bitcoin [12], yang mana digunakan pada sistem pembayaran elektronik dengan arsitektur peer-to-peer yang sangat terdesentralisasi dan tidak memiliki lembaga keuangan untuk bertindak sebagai pengelola dalam transaksi. Blockchain adalah jenis paling dasar dari *Distributed Ledger Technology* (DLT), yang memiliki sifat terdesentralisasi dan menggunakan protokol kesepakatan yang digunakan untuk mencapai consensus.

B. *Smart Contract*

Smart contract merupakan sebuah program kecil yang ada di dalam *blockchain* yang sudah terprogram dan akan berjalan secara otomatis jika syarat kondisi yang sudah ditetapkan terpenuhi [13]. Saat melakukan transaksi, setiap *node* yang terlibat dalam jaringan akan ikut mengeksekusi *smart contract*. Maka dari itu, setiap *node* di dalam *blockchain* harus menyetujui input, output, dan kondisi yang ada di dalam *smart contract* [14].

Smart contract memungkinan adanya pengembangan yang lebih pada teknologi *blockchain* karena keduanya dibangun menggunakan ekosistem yang sama, yaitu dengan jaringan terdesentralisasi. *Blockchain* yang pada awalnya hanya digunakan untuk melakukan proses komputasi sederhana, seperti pencatatan data transaksi, tetapi sekarang telah dikembangkan untuk melakukan proses komputasi yang lebih kompleks dengan integrasi *smart contract* [15]. Kombinasi dari teknologi *blockchain* dan *smart contract* ini dinamakan *Decentralized Application* (DApp). Arsitektur *decentralized application* dapat dilihat pada Gambar 1.



Gambar. 1 Arsitektur aplikasi *Decentralized*

C. Ethereum

Ethereum adalah bentuk aplikasi teknologi blockchain yang paling popular seperti bitcoin. Ethereum dikembangkan untuk melaksanakan tugas komputasi yang lebih kompleks pada kerangka kerja blockchain disbanding hanya komputasi sederhana untuk transaksi data. Serupa dengan bitcoin, ethereum adalah sistem pembayaran mata uang digital terdesentralisasi, menurut definisi resminya. Perbedaan utama dalam Ethereum adalah bahwa sistem ini dikembangkan menggunakan bahasa pemrograman lengkap Turing [6], sehingga memungkinkan pengerjaan komputasi yang lebih kompleks, seperti *smart contract*. Ethereum telah dikenal luas sebagai *framework* dalam mengembangkan *decentralized application*.

D. *InterPlanetary File System*

InterPlanetary File System atau dikenal sebagai IPFS merupakan protokol *hypermedia peer-to-peer* terdistribusi yang digunakan untuk menyimpan dan berbagi konten digital dalam sistem yang terdesentralisasi [9]. IPFS menggunakan *content based addressing* untuk melakukan identifikasi terhadap *file* yang dituju. Hal ini menyebabkan ketika ada perubahan data pada *file* maka akan menghasilkan hasil *hash* yang berbeda juga. IPFS sudah dilengkapi dengan metode *cryptocurrency* untuk *relending*

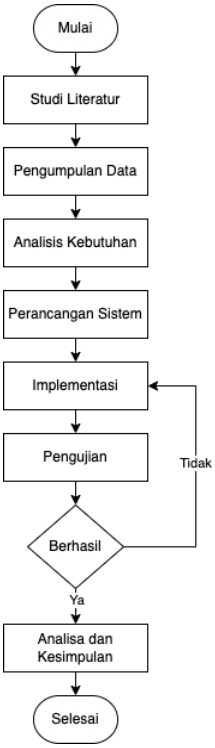
data. Sebagian *user* menggunakan IPFS untuk membagikan *file* berukuran besar karena IPFS menggunakan *hosting local* yang dapat mengurangi kebutuhan *bandwidth* untuk berbagi *file* besar di internet [16]. *File* yang sudah tersimpan di IPFS menghasilkan *hash file* yang kemudian disimpan pada jaringan ethereum. *Hash file* yang tersimpan pada ethereum dapat digunakan untuk mengakses *file* yang tersimpan pada IPFS [17].

E. *Crowdfunding*

Crowdfunding atau biasa disebut dengan *crowd financing*, *equity crowdfunding* atau *hyper funding* merupakan praktik penggalangan dana dari seseorang atau kelompok orang untuk mendukung sebuah proyek atau bisnis yang biasanya dilakukan melalui media internet [18]. Penggalangan dana yang dimaksud seperti penggalangan dana untuk penanggulangan bencana alam, perusahaan *startup* bahkan untuk penelitian ilmiah.

III. METODE PENELITIAN

Metode penelitian merupakan sebuah proses atau langkah-langkah yang dilakukan dalam penelitian. Metode penelitian yang digunakan pada penelitian ini dapat dilihat pada Gambar 2.

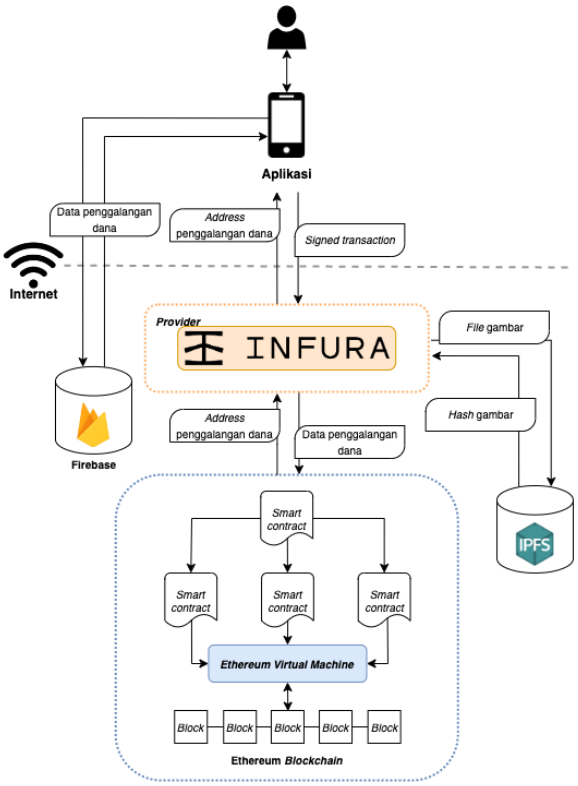


Gambar. 2 *Flowchart* Penelitian

IV. PERANCANGAN

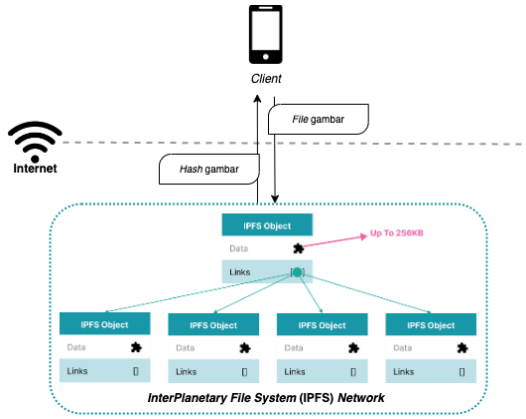
A. *Perancangan Arsitektur Sistem*

Penelitian yang dilakukan adalah membangun sistem penggalangan dana dengan menggunakan *smart contract* ethereum dan *InterPlanetary File System* (IPFS). Sistem yang dibuat kemudian akan diuji untuk membuat penggalangan dana baru yang disimpan pada *smart contract* ethereum. Pengujian pembuatan penggalangan dana ini bertujuan untuk mengetahui *gas fee* yang diperlukan untuk membuat penggalangan dana itu sendiri. Selain menggunakan *smart contract* ethereum, penelitian ini juga memanfaatkan teknologi IPFS sebagai penyimpanan gambar dari penggalangan dana yang dibuat. Penerapan IPFS sendiri yaitu untuk mengurangi beban *smart contract* ethereum dalam menyimpan data gambar penggalangan dana. Pada penelitian ini juga memanfaatkan Firebase sebagai *backup* penyimpanan data penggalangan dana. Detail dari arsitektur sistem yang akan dibangun dapat dilihat pada Gambar 3.



Gambar. 3 Arsitektur sistem yang menerapkan IPFS

Pada Gambar 3 menggambarkan arsitektur sistem penggalangan dana yang menerapkan *smart contract* ethereum dan IPFS. Sistem yang menerapkan IPFS ini divisualisasikan dalam bentuk aplikasi *mobile android*. Cara kerja sistem ini dimulai ketika pengguna akan mengunggah kampanye penggalangan dana ke *smart contract* ethereum. Kemudian ketika pengunggahan dilakukan, sistem akan menandatangani transaksi pengunggahan data kampanye penggalangan dana menggunakan *private key* dari *wallet* pengguna. Ketika proses penandatanganan transaksi selesai, sistem akan mulai mengunggah data kampanye penggalangan dana. Proses pengunggahan yang pertama dilakukan yaitu sistem akan mengunggah gambar terlebih dahulu ke IPFS melalui *infura*. Detail proses penyimpanan data gambar pada IPFS dapat dilihat pada Gambar 4.

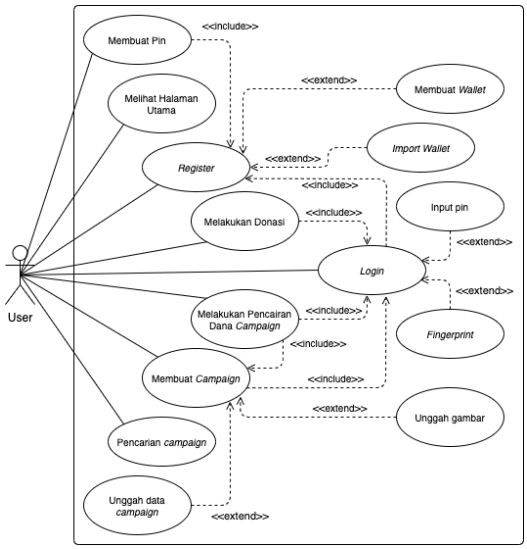


Gambar. 4 Arsitektur sistem IPFS

IPFS menerima inputan *file* gambar. Kemudian *file* gambar tersebut akan disimpan menjadi *object-object* kecil. Pada masing-masing *object* menyimpan data dengan ukuran maksimal 256 KB dan juga menyimpan *link* untuk *object-object* yang lain. Kemudian *object-object* tersebut akan disebarkan ke seluruh *node* pada jaringan. Balikan dari penyimpanan gambar ke IPFS merupakan *hash* gambar. *Hash* ini sendiri berupa *string* dengan panjang 46 karakter yang digunakan sebagai alamat untuk mengakses gambar pada jaringan IPFS. Kemudian *hash* gambar dan data penggalangan dana lainnya akan diunggah bersama ke *smart contract* ethereum menggunakan *infura*. Ketika proses pengunggahan ini berhasil, maka akan mengembalikan *address* penggalangan dana yang diunggah. Sehingga *address* ini yang nantinya akan digunakan sebagai pengenal dari penggalangan dana yang dibuat.

B. Perancangan Perangkat Lunak

Perancangan aktivitas-aktivitas pada sistem digambarkan menggunakan *Unified Modelling Language* (UML). Perancangan sistem pada penelitian ini meliputi diagram *use case* dan deskripsi *use case*.



Gambar. 5 Diagram *Use Case*

Pada Gambar 5 merupakan *use case* diagram aplikasi penggalangan dana yang terdiri dari satu jenis aktor yaitu *user*. *User* dapat melihat halaman utama, mencari *campaign*, melakukan *login* dengan *pin* ataupun *fingerprint*, *register*, donasi, membuat *campaign* dan melakukan pencairan dana *campaign* yang telah terkumpul.

C. Perancangan Basis Data

Pada penelitian ini data yang diperlukan oleh sistem yaitu data konten berupa teks dan gambar. Basis data yang digunakan pada penelitian ini adalah *smart contract* ethereum dan basis data NoSql dengan *firebase*.

TABEL I
RANCANGAN *SMART CONTRACT CROWDFUNDING*

Atribut	Tipe Data	Keterangan
Campaigns	*List of Campaign*	Berisi *object smart contract campaign*

Pada Tabel 1 dapat dilihat rancangan *smart contract crowdfunding*. *Smart contract crowdfunding* menyimpan kumpulan data *campaign* yang sudah dibuat ke dalam *array*.

TABEL II
RANCANGAN *SMART CONTRACT CROWDFUNDING*

Atribut	Tipe Data	Keterangan
image	String	Berisi *hash* gambar
title	String	Berisi judul penggalangan dana
description	String	Berisi deskripsi penggalangan dana
balance	Uint256	Berisi total donasi yang sudah terkumpul
Target	Uint256	Berisi target penggalangan dana yang dibutuhkan
endDate	Uint256	Berisi tanggal penggalangan dana selesai
isComplete	Boolean	Berisi status penggalangan dana apakah sudah selesai atau belum
creatorAddress	Address	Berisi *address wallet* pembuat penggalangan dana
contributors	List of Contributor	Berisi kumpulan data *address wallet* dan jumlah donasi dari setiap donatur

Pada Tabel 2 dapat dilihat rancangan *smart contract campaign*. *Smart contract campaign* digunakan untuk menampung detail data penggalangan dana. Pada *smart contract campaign* terdapat beberapa atribut yaitu *image* untuk menyimpan *hash* gambar, *title* digunakan untuk judul penggalangan dana, *description* digunakan untuk

deskripsi dari penggalangan dana, *balance* digunakan untuk menyimpan total dana penggalangan dana yang terkumpul, target digunakan untuk total dana penggalangan dana yang dibutuhkan, *endDate* untuk waktu berakhirnya penggalangan dana, *isComplete* untuk status penggalangan dana, *creatorAddress* untuk alamat dompet pembuat penggalangan dana, dan *contributors* digunakan untuk menyimpan kumpulan data alamat dompet dan total donasi dari para donatur.

TABEL III
RANCANGAN *COLLECTION CAMPAIGN*

Atribut	Tipe Data	Keterangan
image	String	Berisi *hash* gambar penggalangan dana
title	String	Berisi judul penggalangan dana
description	String	Berisi deskripsi penggalangan dana
target	Number	Berisi total dana penggalangan dana yang dibutuhkan
startDate	Number	Berisi waktu penggalangan dana dimulai
endDate	Number	Berisi waktu penggalangan dana selesai
creatorAddress	String	Berisi alamat dompet pembuat penggalangan dana
transactionHash	String	Berisi *hash* transaksi pembuatan penggalangan dana

Pada Tabel 3 dapat dilihat rancangan *collection campaign*. Data yang disimpan pada *collection campaign* ini hampir sama dengan data yang disimpan pada *smart contract campaign*, bedanya pada *collection campaign* menyimpan *hash* transaksi dari pembuatan penggalangan dana sedangkan pada *smart contract campaign* tidak. Data pada *collection* ini nantinya akan digunakan untuk menyimpan data penggalangan dana, karena proses pencatatan data penggalangan dana ke *smart contract crowdfunding* yang tidak sebentar dan juga jika pembuatan penggalangan dana gagal maka tidak akan tersimpan pada *smart contract* sehingga diperlukan data *backup* untuk *history* data pembuatan penggalangan dana. Selain itu, atribut *transactionHash* digunakan untuk menyimpan *hash* transaksi pembuatan penggalangan dana yang akan digunakan untuk pengecekan status pencatatan data penggalangan dana pada *smart contract*.

TABEL IV
RANCANGAN *COLLECTION HISTORY*

Atribut	Tipe Data	Keterangan
campaignTitle	String	Berisi judul penggalangan dana
category	Number	Berisi kategori transaksi
date	Number	Berisi tanggal transaksi dilakukan
amount	Number	Berisi jumlah ETH yang didonasikan
transactionHash	String	Berisi *hash* transaksi

Pada Tabel 4 dapat dilihat rancangan *collection history*. *Collection history* digunakan untuk menampung data riwayat transaksi. Pada *collection history* terdapat beberapa atribut yaitu *campaignTitle* digunakan untuk menyimpan judul penggalangan dana, *category* digunakan untuk kategori transaksi yang dilakukan (*category* 1 untuk pembuatan *campaign*, *category* 2 untuk donasi dan *category* 3 untuk pencairan penggalangan dana), *amount* digunakan untuk jumlah ETH yang didonasikan dan *transactionHash* digunakan untuk *hash* transaksi.

V. HASIL DAN PEMBAHASAN

A. *Kode Program*

Pada penelitian ini perangkat lunak dibangun menggunakan *framework Hardhat* dan *Flutter*. *Framework hardhat* dibuat dengan bahasa pemrograman *Javascript* yang bisa menjalankan bahasa pemrograman *Solidity* untuk membuat *smart contract* ethereum. Kode program *smart contract* pada penelitian ini dibuat dengan teknik *multi-contract*. *Multi-contract* merupakan pembagian *smart contract* menjadi *contract-contract* kecil, yang mana nantinya *contract-contract* ini akan saling terhubung satu sama lain. Pada penelitian ini *smart contract* yang dibuat dibagi menjadi *smart contract crowdfunding* dan *smart contract campaign*.

Smart contract crowdfunding merupakan *smart contract* utama yang diakses untuk pembuatan *campaign*. Pada *smart contract* ini terdapat fungsi untuk pembuatan *campaign* dan mendapatkan data *address campaign*.

```
function createCampaign(
    string memory _image,
    string memory _title,
    string memory _description,
    uint256 _target,
    uint256 _endDate
) public {
    require(_target > 0);

    address newCampaign = new Campaign(
        _image,
        _title,
        _description,
        _target,
        _endDate,
        msg.sender
    );
    campaigns.push(newCampaign);
}
```

Gambar. 6 Kode program *Smart Contract* pembuatan *Campaign*

Pada gambar 6 merupakan kode program untuk pembuatan *campaign*. Pembuatan *campaign* memerlukan beberapa data yaitu *image*, *title*, *description*, *target* dan *endDate*. Semua data ini wajib diisi agar proses pembuatan *campaign* berhasil. Terdapat pengecekan data *target* yang dimasukan. Data target yang dimasukan merupakan *target* dana *campaign* yang diharapkan, jika target yang dimasukan lebih dari 0 maka akan dilanjutkan ke proses pembuatan *campaign*. Kemudian dilakukan proses pembuatan *campaign* ke *smart contract campaign*. Ketika proses pembuatan *campaign* berhasil maka akan mengembalikan *address campaign* yang kemudian disimpan pada *array campaigns*.

Smart contract campaign merupakan tempat untuk menyimpan data *campaign*. Selain untuk menyimpan data *campaign*, pada *smart contract* ini juga terdapat fungsi untuk melakukan donasi *campaign* dan fungsi untuk *claim* dana *campaign*.

```
contract Campaign {
    string public image;
    string public title;
    string public description;
    uint256 public balance;
    uint256 public target;
    uint256 public endDate;
    bool public isComplete;
    address public creatorAddress;
    Contribute[] public contributors;

    function Campaign(
        string memory _image,
        string memory _title,
        string memory _description,
        uint256 _target,
        uint256 _endDate,
        address _creatorAddress
    ) public {
        image = _image;
        title = _title;
        description = _description;
        target = _target;
        endDate = _endDate;
        creatorAddress = _creatorAddress;
    }
}
```

Gambar. 7 Kode Program *Smart Contract Constructor Campaign*

Pada gambar 7 merupakan kode program *constructor* untuk menyimpan data *campaign*. *Constructor* merupakan perintah khusus yang akan dieksekusi pertama kali ketika pembuatan *object* (*instance*). Data *campaign* yang disimpan ketika proses pembuatan *object* yaitu *image*, *title*, *description*, *target*, *endDate* dan *creatorAddress*. Data *image*, *title*, *description*, *target* dan *endDate* merupakan data yang dimasukkan dari *user* ketika membuat *campaign*. Sedangkan data *creatorAddress* merupakan *address wallet user* yang membuat *campaign*.

```
function contribute() public payable {
    require(isComplete == false);
    require(now <= endDate);
    require(msg.value > 0);
    require(balance < target);

    Contribute memory newContribute = Contribute({
        amount: msg.value,
        contributor: msg.sender
    });

    contributors.push(newContribute);
    balance += msg.value;
}
```

Gambar. 8 Kode program *Smart Contract* kontribusi donasi

Pada gambar 8 merupakan kode program untuk melakukan donasi. Proses donasi ini memerlukan masukan berupa jumlah ETH yang ingin didonasikan. Jumlah ETH ini dikirim melalui *variabel global* yaitu *msg.value* yang terdapat pada bahasa pemrograman solidity. Terdapat beberapa pengecekan sebelum donasi dapat dijalankan, yaitu status *campaign*, tanggal berakhir *campaign*, jumlah dana donasi yang dimasukkan dan jumlah dana donasi yang sudah terkumpul. Terdapat juga proses pembuatan *riwayat contributor* berupa jumlah donasi dan *address wallet* donatur yang disimpan pada tipe data *struct* "*Contribute*".

```
modifier onlyCreator() {
    require(msg.sender == creatorAddress);
    _;
}

function deliverBalance() public payable onlyCreator {
    require(isComplete == false);
    require(balance > 0);

    creatorAddress.transfer(balance);
    isComplete = true;
}
```

Gambar. 9 Kode Program *Smart Contract Claim* Dana Donasi

Pada gambar 9 merupakan kode program untuk fungsi *claim* dana donasi. Fungsi ini dibuat dengan tipe *payable* yang artinya di dalam fungsi ini terdapat proses untuk mengirim dana ke *address wallet* lain. Selain itu, pada fungsi ini juga terdapat beberapa pengecekan untuk *address wallet* yang melakukan *claim* dana donasi, status *campaign* dan juga jumlah donasi yang terkumpul. Ketika semua pengecekan berhasil maka dana donasi yang terkumpul akan ditransfer ke *address wallet* yang melakukan *claim* dana dan status *campaign* akan berubah menjadi selesai.

B. Pengujian Sistem

Pengujian biaya transaksi *smart contract* dengan IPFS dilakukan dengan mengunggah kampanye penggalangan dana ke *smart contract* ethereum menggunakan sistem yang telah dibuat. Data kampanye penggalangan dana yang digunakan untuk pengujian dikumpulkan melalui website www.kitabisa.com seperti yang terlampir pada Lampiran.

Pada pengujian ini, data kampanye penggalangan dana diunggah ke *contract* yang telah dibuat pada *smart contract* ethereum dengan *address* 0x2acb457e294cc18c1162ed3d43bd03253d1f5252. Setelah pengunggahan *campaign* berhasil, data detail dari proses pengunggahan seperti biaya transaksi, kecepatan transaksi dan status transaksi dilihat melalui website https://goerli.etherscan.io berdasarkan *hash transaction* dari pengunggahan kampanye penggalangan dana. Hasil pengujian dapat dilihat pada Tabel 5.

TABEL V
PENGUJIAN *GAS SMART CONTRACT* DENGAN IPFS

Pengujian	Biaya Transaksi (ETH)	Biaya Transaksi (Rupiah)	Kecepatan Transaksi (Detik)
1	0,00344645	69.562	36
2	0,00340337	68.692	12
3	0,003289158	66.387	24
4	0,00340343	68.693	12
5	0,003460778	69.851	12
6	0,003289217	66.388	24
7	0,003459848	69.832	24
8	0,003289098	66.386	24
9	0,003460718	69.850	12
10	0,003460268	69.841	24
11	0,00340322	68.689	36
12	0,00340322	68.689	24
13	0,003232233	65.238	36
14	0,003460628	69.848	12
15	0,00357394	72.135	12
16	0,00340373	68.699	24
17	0,003346625	67.547	24
18	0,003232203	65.237	12
19	0,003232783	65.249	24
20	0,003346445	67.543	12
21	0,0035743	72.142	24
22	0,003346565	67.546	12
23	0,00340328	68.690	24
24	0,003175945	64.102	36
25	0,00340328	68.690	12
26	0,00340328	68.690	12
27	0,00340343	68.693	24
28	0,003289378	66.391	36
29	0,003232413	65.242	12
30	0,003232023	65.234	24
31	0,003288858	66.381	12
32	0,003460388	69.843	12
33	0,003460028	69.836	12
34	0,00340295	68.684	24
35	0,003232833	65.250	12
36	0,00311847	62.942	36
37	0,003346355	67.541	24
38	0,0034032	68.689	12
39	0,00340346	68.694	12
40	0,003232383	65.241	12

Berdasarkan Tabel 5 *gas price* yang ditetapkan untuk semua proses pengunggahan kampanye penggalangan dana yaitu 0,0000000025 ETH. Sedangkan untuk *max gas price* yang digunakan yaitu 30.000 *units*. Berdasarkan pengujian yang dilakukan biaya transaksi terbesar terjadi ketika pengujian ke 15 dengan total biaya 0,00357394 ETH, sehingga rata-rata biaya transaksi yang diperlukan yaitu sebesar 0,003360254 ETH. Transaksi tercepat yang berhasil dilakukan pada penelitian ini yaitu 12 detik sedangkan yang terlama yaitu 36 detik dengan rata-rata kecepatan transaksi yaitu 21 detik.

Pengujian biaya transaksi *smart contract* tanpa IPFS dilakukan dengan mengunggah kampanye penggalangan dana menggunakan *code editor* online yaitu https://remix.ethereum.org. Perbedaan antara pengujian tanpa dan dengan IPFS yaitu terletak pada proses pengunggahan gambar. Pada pengujian tanpa IPFS, gambar diunggah dengan format base64, sehingga sebelum diunggah gambar perlu dikonversi ke bentuk base64 terlebih dahulu. Proses konversi ini mempengaruhi ukuran gambar asli gambar yang membuat ukuran gambar menjadi lebih besar. Pada pengujian ini, kampanye penggalangan dana diunggah ke *smart contract* dengan *address* 0x4ee9aa39f40526f342725e77a6fbb9188319d4e9 pada jaringan goerli ethereum. Hasil pengujian dapat dilihat pada Tabel 6.

TABEL VI
PENGUJIAN *GAS SMART CONTRACT* TANPA IPFS

Pengujian	Biaya Transaksi (ETH)	Biaya Transaksi (Rupiah)	Kecepatan Transaksi (Detik)
1	0,06216976	1.254.808	12
2	0,060246948	1.215.999	12
3	0,037799195	762.923	12
4	0,040932105	826.156	24
5	0,058309805	1.176.900	12
6	0,049192738	992.885	24
7	0,040362385	814.657	12
8	0,036659185	739.913	48
9	0,069250508	1.397.723	12
10	0,06480517	1.308.000	12
11	0,051301128	1.035.440	48
12	0,05893618	1.189.543	12
13	0,051585673	1.041.183	24
14	0,041786873	843.408	24
15	0,061214728	1.235.532	36
16	0,062982258	1.271.207	12
17	0,042242338	852.601	12
18	0,059732965	1.205.625	12
19	0,056372403	1.137.796	12
20	0,05796693	1.169.980	24
21	0,062697182	1.265.453	12
22	0,06816799	1.375.873	12
23	0,063209959	1.275.803	24
24	0,072214617	1.457.549	12
25	0,05073095	1.023.932	12
26	0,065945228	1.331.010	24
27	0,056747662	1.145.370	12
28	0,062031147	1.252.010	12
29	0,067841485	1.369.283	12
30	0,074473018	1.503.131	12
31	0,061157983	1.234.386	12
32	0,07175852	1.448.343	12

Pengujian	Biaya Transaksi (ETH)	Biaya Transaksi (Rupiah)	Kecepatan Transaksi (Detik)
33	0,072669446	1.466.729	12
34	0,073353341	1.480.532	12
35	0,073297006	1.479.395	12
36	0,060634499	1.223.821	12
37	0,081578707	1.646.550	12
38	0,065324124	1.318.474	60
39	0,061627179	1.243.857	12
40	0,070761949	1.428.229	12

Berdasarkan Tabel 5.3 biaya transaksi terbesar yaitu terdapat pada pengujian ke 37 dengan total 0,08157870 ETH. Rata-rata biaya transaksi pada penelitian ini yaitu 0,060001782 ETH. Transaksi tercepat yang berhasil dilakukan pada penelitian ini yaitu 12 detik sedangkan yang terlama yaitu 60 detik dengan rata-rata kecepatan transaksi yaitu 19,8 detik. Sama seperti pengujian dengan sistem yang menerapkan IPFS, pada sistem yang tidak menerapkan IPFS ini juga menggunakan *gas price* yaitu 0,0000000025 ETH dan *max gas price* yaitu 30.000 *units*.

C. Pembahasan

Penelitian ini bertujuan untuk membangun aplikasi penggalangan dana terdesentralisasi yang menggunakan jaringan Ethereum dan *InterPlanetary File System* (IPFS) untuk mengefisienkan penggunaan *Gas Fee Smart Contract*. Implementasi IPFS digunakan sebagai penyimpanan desentralisasi untuk gambar dari kampanye penggalangan dana yang dibuat. Sistem yang dibangun pada penelitian ini ada 2 yaitu sistem penggalangan dana yang menerapkan IPFS dan yang tidak menerapkan IPFS. Pengujian untuk sistem yang menerapkan IPFS dilakukan dengan aplikasi *android* sedangkan pengujian sistem yang tidak menerapkan IPFS dilakukan dengan mengakses *code editor online* yaitu https://remix.ethereum.org. Hasil pengujian pengunggahan kampanye penggalangan dana didapatkan melalui situs https://goerli.etherscan.io berdasarkan *transaction hash* pada proses pengunggahan kampanye penggalangan dana yang dilakukan.

Penggunaan IPFS ternyata berhasil mengurangi *gas fee smart contract* untuk transaksi pada jaringan ethereum. Hal ini terjadi dikarenakan ukuran besar data yang harus disimpan pada *smart contract* ethereum menjadi lebih kecil. Penggunaan IPFS menyebabkan gambar yang sebelumnya harus disimpan pada *smart contract* ethereum sebagai base64 sekarang hanya perlu menyimpan *hash* dari gambar yang telah disimpan pada IPFS. Persentase penururan pada penelitian ini dihitung dengan persamaan 1.

$$\text{Persentase Penurunan} = \frac{(\text{Nilai Awal} - \text{Nilai Akhir})\ x\ 100\%}{\text{Nilai Awal}} \quad (1)$$

Pengujian pertama dilakukan dengan menggunakan jaringan rinkeby ethereum. Hasil dari pengujian ini rata-rata biaya transaksi turun 93,64% begitu juga dengan kecepatan transaksi yang turun 1,96%. Pengujian ini dilakukan sebelum ethereum *merge* yaitu pada saat masih menggunakan *consensus Proof-of-Work*. Sehingga ketika *merge* selesai, jaringan ini sudah tidak bisa digunakan dan dialihkan untuk menggunakan jaringan goerli yang menerapkan *consensus* baru yaitu *Proof-of-Stake*. Kemudian dilakukan pengujian ulang menggunakan jaringan goerli ethereum.

Pengujian menggunakan jaringan goerli ethereum pada sistem yang menerapkan IPFS dengan total 40 data terbukti dapat menurunkan *gas fee smart contract* ethereum dengan rata-rata penurunan 94,39%. Berbeda dengan biaya *gas fee* yang menurun, kecepatan validasi transaksi *smart contract* ethereum pada jaringan ini untuk sistem yang menerapkan IPFS cenderung lebih tinggi dengan rata-rata peningkatan 13,55%. Perbandingan *gas fee* dan kecepatan validasi transaksi dari hasil pengujian dengan jaringan goerli ethereum dapat dilihat pada Tabel 7.

TABEL VII
PERBANDINGAN RATA-RATA *GAS FEE SMART CONTRACT*

Rata-Rata *Gas Fee* dan Kecepatan Validasi Transaksi *Smart Contract* Dengan IPFS		**Rata-Rata *Gas Fee* dan Kecepatan Transaksi *Smart Contract* Tanpa IPFS**	
Kecepatan Transaksi (Detik)	**Biaya Transaksi (ETH)**	**Kecepatan Transaksi (Detik)**	**Biaya Transaksi (ETH)**
20,1	0,003360254	17,7	0,060001782
Persentase kenaikan kecepatan validasi transaksi			13,55%
Persentase penurunan *gas fee*			94,39%

VI. KESIMPULAN

Berdasarkan hasil penelitian yang telah diperoleh, maka dapat disimpulkan bahwa Penerapan IPFS pada sistem penggalangan dana untuk penyimpanan data kampanye membutuhkan rata-rata *gas fee* 0,00311847-0,003379868 ETH dengan waktu validasi transaksi rata-rata 21 detik. Berdasarkan pengujian sebanyak 40 kali dengan data yang berbeda, penerapan IPFS dapat menurunkan *gas fee* dengan rata-rata hingga 94,39% dan kecepatan transaksi sistem yang menerapkan IPFS lebih besar 13,55% dari sistem yang tidak menerapkan IPFS. Hal ini dikarenakan data gambar kampanye penggalangan dana tidak perlu disimpan langsung pada *smart contract* ethereum, sehingga data kampanye penggalangan dana yang harus disimpan pada *smart contract* ethereum menjadi lebih kecil dan mengurangi kompleksitas transaksi

REFERENSI

[1] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," International Journal of Network Security, vol. 19, no. 5, pp. 653–659, Sep. 2017, doi: 10.6633/IJNS.201709.19(5).01.

[2] A. Fauzan N I, "TEKNOLOGI BLOCKCHAINDAN PERANANNYA DALAM ERA DIGITAL," Jurnal BJB University, vol. 4, pp. 1–15, Dec. 2018.

[3] H. A. Mutar and M. S. Al-Huseiny, "Implementation of national cryptocurrency using ethereum development platform," in Proc. IUS Conference, vol. 7, no. 3, pp. 1021–1029, 2019. [Online]. Available: http://pen.ius.edu.ba

[4] G. Hileman and M. Rauchs, "2017 Global Blockchain Benchmarking Study," SSRN Electron. J., 2017, doi: 10.2139/ssrn.3040224.
[5] G. Hardeman, "Replacing Paper Contracts With Ethereum Smart Contracts," U.S. Patent Pending, 2016.
[6] V. Buterin, "A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM," Webpage, 2014. [Online]. Available: [Link not provided].
[7] D. D. Wood, "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER," Computer Science Report, 2014.
[8] M. Ainun Fajar, "QUALITY OF SERVICE ETHEREUM BLOCKCHAIN BERBASIS IPFS UNTUK VALIDASI IJAZAH SEKOLAH," Article, 2020.
[9] N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," in Lecture Notes in Computer Science, vol. 10974 LNCS, pp. 199–212, 2018, doi: 10.1007/978-3-319-94478-4_14.
[10] M. N. Saadat, S. A. H. S. A. Rahman, R. M. Nassr, and M. F. Zuhiri, "Blockchain based crowdfunding systems in Malaysian perspective," in Proc. PervasiveHealth Conference, pp. 57–61, Feb. 2019, doi: 10.1145/3313991.3313999.
[11] A. Hoffman, P. Austria, C. H. Park, and Y. Kim, "Bountychain: Toward Decentralizing a Bug Bounty Program with Blockchain and IPFS," International Journal of Networked and Distributed Computing, vol. 9, no. 2–3, pp. 86–93, Jun. 2021, doi: 10.2991/IJNDC.K.210527.001.
[12] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Whitepaper, 2008.
[13] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?," Future Internet, vol. 10, no. 2, Feb. 2018, doi: 10.3390/FI10020020.
[14] R. Azzi, R. K. Chamoun, and M. Sokhn, "The power of a blockchain-based supply chain," Comput Ind Eng, vol. 135, pp. 582–592, Sep. 2019, doi: 10.1016/J.CIE.2019.06.042.
[15] S. Jani, "Smart Contracts: Building Blocks for Digital Transformation," Tech. Report, 2020, doi: 10.13140/RG.2.2.33316.83847.
[16] E. Nyaletey, R. M. Parizi, Q. Zhang, and K. K. R. Choo, "BlockIPFS - Blockchain-enabled interplanetary file system for forensic and trusted data traceability," in Proc. IEEE International Conference on Blockchain, pp. 18–25, Jul. 2019, doi: 10.1109/BLOCKCHAIN.2019.00012.
[17] A. Rajalakshmi, Sindhu, and A. Amritha, "A Blockchain and IPFS based framework for secure Research record keeping," Tech. Report, 2018.
[18] E. Mollick, "The dynamics of crowdfunding: An exploratory study," J Bus Ventur, vol. 29, no. 1, pp. 1–16, Jan. 2014, doi: 10.1016/J.JBUSVENT.2013.06.005.