Implementasi Ethereum Claims Registry pada Ethereum Blockchain untuk Verifikasi Transaksi Konten Fotografi dengan InterPlanetary File System

Reyner, Rudy Adipranata, Leo Willyanto Santoso
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jln. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031)-2983455, Fax. (031)-8417658
E-mail : reynerbpantou@gmail.com, rudya@petra.ac.id, leow@petra.ac.id

ABSTRAK

Pada zaman sekarang fotografi kerap mendapatkan banyak perhatian dari berbagai kalangan. Fotografi dapat digunakan untuk merekam momen-momen penting maupun lalu diabadikan dalam format digital dan disimpan ke sebuah penyimpanan. Pada umumnya konten fotografi disimpan ke *server*, namun terdapat ancaman pada penyimpanan terpusat seperti serangan *malware*, malfungsi pada *hardware*, dan *human error*.

Pada penelitian ini akan dilakukan implementasi Ethereum Claims Registry pada smart contract untuk verifikasi atribut tertentu suatu user yang tersimpan di Ethereum Blockchain, sehingga user dapat memberikan atau mengambil claim yang ada. Lalu implementasi InterPlanetary File System sebagai penyimpanan website dan konten fotografi. Website dan fotografi yang tersimpan pada *InterPlanetary File System* dapat diakses secara global tanpa penyimpanan terpusat dan tidak dikenakan biaya tambahan pada transaksi *ethereum* karena disimpan *off-chain* atau diluar *blockchain*.

Berdasarkan hasil pengujian yang telah dilakukan, implementasi *Ethereum Claims Registry* dan *InterPlanetary File System* sebagai penyimpanan konten fotografi untuk *high gas price* membutuhkan 0.020518 ETH, *medium gas price* 0.010747 ETH, dan *low gas price* 0.004885 ETH. Lalu implementasi *Ethereum Claims Registry* memperoleh efisiensi *ether* sebanyak 1% pada *deploy smart contract*, -0.4% pada *make claim*, dan -0.4% pada *set claim*.

Kata Kunci: *blockchain, Ethereum Claims Registry, Interplanetary File System.*

ABSTRACT

Nowadays, photography often receives a lot of attention from diverse backgrounds. Photography can be used to capture important moments as well as then digitally encased and stored into a storage unit. Typically, photography content is stored on servers, but there are threats in centralized storage like malware attacks, hardware failures and human errors.

In the study, Ethereum Claims Registry will be implemented on smart contracts to verify certain attributes about its user which are stored in Ethereum Blockchain, so that users can either issue or retrieve existing claims. Then implementation of the interplanetary file system as a website storage and photography content. Website and Photography Content which are stored on InterPlanetary File System can be accessed globally without centralized storage and do not incur additional fees on ethereum transactions because they are stored off-chain or stored outside the blockchain.

Based on the results of the tests that have been carried out, the implementation of the Ethereum Claims Registry and the InterPlanetary File System as a storage for photography content for high gas price requires 0.020518 ETH, medium gas price requires 0.010747 ETH, and low gas requires prices 0.004885 ETH. Then the implementation of the Ethereum Claims Registry obtained 1% ether efficiency on deploying smart contracts, -0.4% on make claims, and -0.4% on set claims.

Keywords: *blockchain, Ethereum Claims Registry, Interplanetary File System.*

1. PENDAHULUAN

Pada zaman sekarang fotografi kerap menjadi sorotan atau mencuri banyak perhatian banyak kalangan. Kata fotografi merupakan kata serapan yang Berasal dari bahasa Inggris yaitu *photography*. Pengertiannya berasal dari dua kata yaitu photos dan grafo yang berarti cahaya dan melukis. Fotografi merupakan seni menangkap cahaya dengan kamera, biasanya melalui sensor atau film digital, untuk membuat gambar [4]. Dengan peralatan kamera yang tepat, maka bisa memotret gelombang cahaya yang tidak terlihat oleh mata manusia, termasuk *UV*, inframerah, dan radio [4]. Fotografi digunakan untuk merekam momen-momen penting maupun membagikan Informasi lalu diabadikan menjadi sebuah format digital.

Setiap hasil foto harus disimpan pada suatu media penyimpanan. Media penyimpanan fotografi tentu memiliki kerentanan yang disebabkan oleh aspek teknologi atau kesalahan manusia. Diantaranya adalah pertama, *file* foto yang rentan terserang *malware*. Dunia IT sangat terkenal dengan virus terutama *ransomware*, virus ini sangat merugikan korban karena dapat menyerang komputer korban dengan meng-enkripsi semua file. Dikutip dari New Indian Express, seorang fotografer dirugikan karena komputernya terserang *ransomware* dan diperas untuk melakukan pembayaran agar dapat decrypt file fotonya [6]. Kedua, faktor *human error*. Manusia tidak terlepas dari kesalahan sebagai contoh kasus lemniscate pada forum reddit di tahun 2017 yang mengalami sebuah insiden yang tidak diinginkan. Kala itu, lemniscate menyewa seorang fotografer untuk acara wedding. Fotografer mengunggah foto lemniscate pada salah satu platform penyimpan foto. Lemniscate baru mengunduh sekitar 10% dari keseluruhan foto, namun secara sepihak fotografer malah menghapus semua foto Lemniscate pada *platform* maupun penyimpanan local [8]. Terakhir, faktor file dicuri. Contoh kasus seperti yang dilansir oleh detikNews bahwa Seorang fotografer *National Geographic* bernama Joel Santore kecurian tas di

Bandara I Gusti Ngurah Rai yang menyebabkan hilangnya ratusan foto spesies-spesies langka yang ada di Indonesia [7].

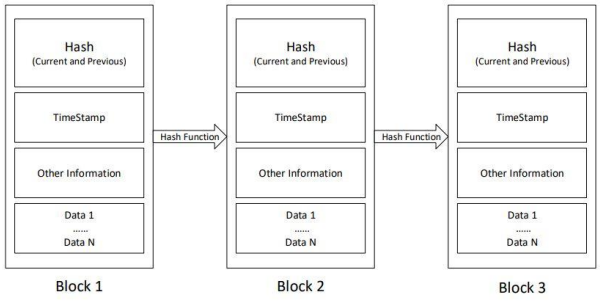
Pada penelitian sebelumnya menggunakan *Ethereum Blockchain* untuk menyimpan Informasi gambar kemudian mengambil *Transaction ID* (*TxHash*) untuk diolah menjadi *watermark* pada gambar [3]. Dengan Implementasi *Blockchain*, data yang disimpan secara terdistribusi dapat membantu mengatasi permasalahan pada *versioning* pada konten gambar, namun gambar yang tetap disimpan pada server terpusat. Salah satu penelitian melakukan kombinasi *Ethereum Blockchain* dan *InterPlanetary File System* untuk mengatasi masalah *versioning* dan sistem yang masih terpusat. Menurut peneliti penyimpanan yang terpusat bisa mengakibatkan *history* pada suatu konten bisa dirusak dan memberikan resiko terhadap kredibilitas sehingga dibutuhkan *platform* yang terdesentralisasi (Nizamuddin et al, 2019).

Sistem yang akan dibuat pada skripsi ini akan mengimplementasikan *Ethereum Claims Registry* dalam transaksi pada *Ethereum Blockchain* untuk proses enkripsi dan verifikasi authority sehingga hanya owner dan user yang ditentukan saja yang dapat melakukan akses terhadap file yang disimpan dan *InterPlanetary File System* untuk menyimpan konten fotografi sekaligus sebagai sarana untuk men-*deploy* website, sehingga platform dapat diakses secara global dan relatif menggunakan biaya (gas) yang lebih kecil.

2. DASAR TEORI

2.1 Blockchain

Blockchain merupakan sebuah buku besar yang terdistribusi dimana *blockchain* terdiri dari *block* yang saling berhubungan dan membentuk pola rantai (*chain*). Semua Transaksi yang terjadi pada jaringan blockchain akan tersimpan kedalam list yang disebut dengan *block*. Setiap *block* memiliki *block number*, *timestamp*, *transaction data*, dan *cryptography hash* dari *previous block* sehingga data yang ada berkesambungan dan membentuk pola rantai seperti konsep dari *Blockchain* [10]. Berbeda dengan *block* lainnya, *block* pertama atau yang biasa disebut dengan *Genesis Block* tidak menggunakan *previous hash* namun menggunakan *hash* pada *block* itu sendiri. Struktur pada blockchain dapat dilihat pada Gambar 1.



Gambar 1. Struktur blockchain [10]

Teknologi *Blockchain* menyediakan sebuah *immutable data storage* yang memungkinkan transaksi hanya ditambahkan dan tidak pernah diubah atau dihapus [11]. *Blockchain* biasanya di-deploy menjadi salah satu dari tiga *environment* atau tipe tergantung dengan *access permissions* yang dibutuhkan, diantaranya [5]:

1. *Public blockchain* : Merupakan salah satu *environment* yang permisif, dimana *environment* ini mengizinkan *public participant* untuk melakukan akses ke sistem. Orang yang terhubung juga dapat melakukan fitur *read* dan *write*.
2. *Consortium blockchain* : Hanya *subset* dari *consensus participant* yang dapat berkontribusi ke dalam protocol untuk menambah block. Untuk fitur read dapat dibatasi hanya untuk consensus participant atau ke public participant.
3. *Private blockchain* : Merupakan salah satu *environment* yang paling tertutup, dimana fitur write hanya diberikan kepada suatu organisasi dan biasanya di-*deploy* pada jaringan *private* pada suatu organisasi.

Tabel 1. Perbandingan antara *Public*, *Consortium*, dan *Private Blockchain* [2]

Item	Public	Consortium	Private
Akses	*Read/Write* untuk keseluruhan	*Read/Write* untuk satu organisasi	Read/Write untuk beberapa organisasi yang dipilih
Kecepatan	Lebih Lambat	Lebih ringan dan lebih cepat	Lebih ringan dan lebih cepat
Efisiensi	Rendah	Tinggi	Tinggi
Keamanan	*Proof of work*, *proof of stake*, dan mekanisme konsensus lainnya	Peserta yang disetujui sebelumnya dan konsensus *multi-party*	Peserta yang disetujui sebelumnya dan konsensus *multi-party*
Kekekalan	Hampir mustahil untuk dirusak	Bisa dirusak	Bisa dirusak
Konsensus	Tanpa izin dan anonim	Dengan izin dan identitas dikenal	Dengan izin dan identitas dikenal

Pada Tabel 1 tertera perbandingan atau perbedaan dari *public blockchain*, *consortium blockchain*, dan *private blockchain*. Masing-masing *environment* menghadapi *challenges* yang berbeda, dimana *consortium blockchain* dan *private blockchain* rentan terkena tamper dan serangan 51% *known attack* namun memiliki performa yang lebih bagus dibanding *public blockchain*. Sedangkan *public* menyajikan *immutability* dan *availability* yang tinggi meski memiliki performa tidak sebagus *environment* yang lain.

2.2 Ethereum Claims Registry

Ethereum Claims Registry merupakan proposal yang diajukan oleh komunitas Ethereum yang berfungsi agar smart contract dan user bisa melakukan *claim* satu sama lain maupun *self-claim* pada transaksi yang dibuat pada jaringan *Ethereum*. *Ethereum Claims Registry* merupakan *guidelined* atau *standard* dalam melakukan *claim* dengan menggunakan bahasa *Solidity* dan ditulis pada *Smart Contract.* Ethereum Claims Registry diterapkan dengan menggunakan kombinasi storage variable dengan mapping kemudian disesuaikan dengan parameter yang dibutuhkan. Contoh code Ethereum Claims Registry dapat dilihat pada Gambar 2.

Appendix: Registry implementation

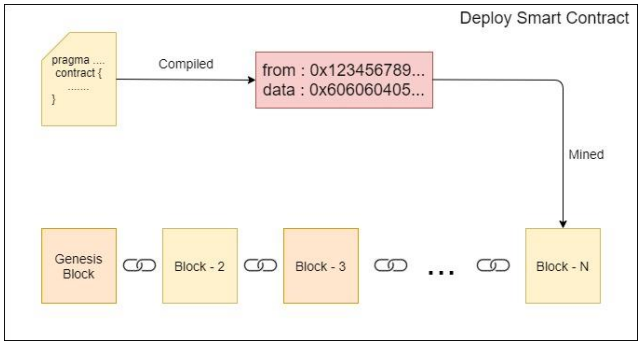
```
contract EthereumClaimsRegistry {

    mapping(address => mapping(address => mapping(bytes32 => bytes32))) public registry;

    event ClaimSet(
        address indexed issuer,
        address indexed subject,
        bytes32 indexed key,
        bytes32 value,
        uint updatedAt);

    event ClaimRemoved(
        address indexed issuer,
        address indexed subject,
        bytes32 indexed key,
        uint removedAt);

    // create or update clams
    function setClaim(address subject, bytes32 key, bytes32 value) public {
        registry[msg.sender][subject][key] = value;
        emit ClaimSet(msg.sender, subject, key, value, now);
    }
```

Gambar 2. *Ethereum Claims Registry*

2.3 Smart Contract

Smart Contract adalah kode yang dapat dieksekusi yang berjalan di blockchain untuk memfasilitasi, mengeksekusi, dan menegakkan ketentuan perjanjian [1]. *Smart Contract* juga bisa dibilang sebagai dokumen berbentuk digital yang mengikat perjanjian atau kesepakatan antara kedua belah pihak. Kontrak yang sudah dibuat bersifat permanen dan tidak dapat dimodifikasi, jika pemilik dari *smart contract* ingin melakukan perubahan terhadap isi kontrak maka pemilik harus melakukan *deploy* ulang *smart contract* tersebut kedalam jaringan blockchain dan smart contract yang baru di-deploy akan mendapatkan *address* yang baru. *Smart contract* bisa digunakan didalam sistem untuk membuat sistem lebih terpercaya dan otomatis. Salah satu *blockchain* yang mendukung smart contract adalah *Ethereum* dengan menggunakan bahasa *Solidity*.

Pada *ethereum blockchain*, smart contract di-compile terlebih dahulu menjadi Ethereum Virtual Machine Bytecode (EVM Bytecode) lalu *compiled code* dikirim ke jaringan blockchain. Namun setelah di-deploy smart contract tidak langsung bisa digunakan, smart contract harus melalui proses *bake (e.g: Mining dan Staking)* sesuai dengan *consensus algorithm* pada *blockchain* yang digunakan. Setelah smart contract sudah di-*mine* atau *stake*, maka *state* pada smart contract berubah menjadi *success* dan *smart contract* bisa digunakan. Detail saat melakukan *deploy smart contract* dapat dilihat pada Gambar 3.



Gambar 3. Deploy Smart Contract

2.4 InterPlanetary File System

InterPlanetary File System atau yang dikenal sebagai IPFS merupakan arsitektur *peer-to-peer* yang digunakan untuk menyimpan dan berbagi data dalam sistem yang terdesentralisasi. *InterPlanetary File System* menggunakan *Content Based Addressing* untuk melakukan identifikasi terhadap *file* yang dituju. Karena menggunakan *Content Based Addressing* maka ketika ada perubahan data, akan berakibat perubahan *hash* pada *Content Based Addressing*. *InterPlanetary File System* juga sudah dilengkapi dengan metode *cryptography* untuk relending data yang terdapat pada sistem *InterPlanetary File System* sendiri. Sebagian besar user menggunakan *IPFS* untuk membagikan file berukuran besar karena *IPFS* menggunakan hosting local yang mengurangi kebutuhan bandwidth yang biasanya dikaitkan dengan berbagi file besar pada internet [9].

3. DESAIN SISTEM

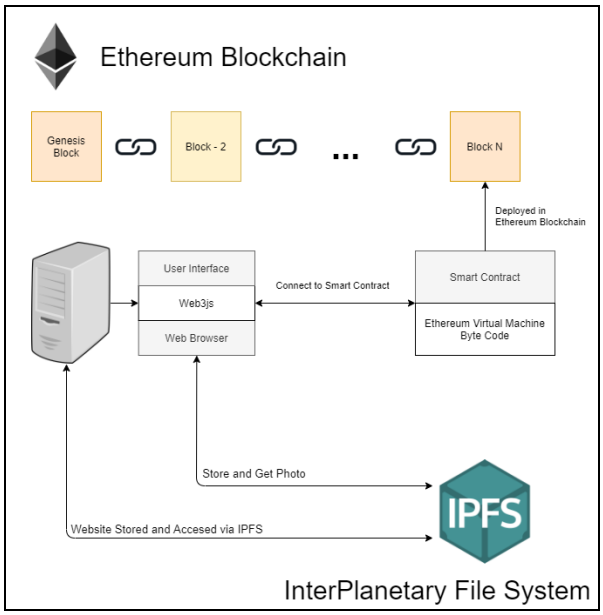
3.1 Analisa Masalah

Penelitian ini dilandasi oleh masalah keamanan khususnya fotografer dalam menyimpan konten fotografi. Masalah serangan *malware* dan *human error* kerap kali menjadi masalah yang sulit dihindarkan pada sebuah sistem yang masih terpusat atau *centralized*.

Solusi yang sudah diberikan pada penelitian serupa sebelumnya yaitu menggunakan *Blockchain* sebagai wadah untuk *versioning* pada konten multimedia sehingga bisa divalidasi keaslian dari suatu konten. Namun solusi yang diberikan belum bisa menjawab *avalaibility* terkait konten, yang berarti meski sudah transaksi sudah tertulis dan tersebar ke jaringan blockchain konten media masih bisa terkena *tamper* atau *hilang* karena *human error*. Oleh karena itu, pengembangan dari kekurangan penelitian sebelumnya dilakukan dengan Implementasi InterPlanetary File System dan Ethereum Claims Registry pada sistem penyimpanan konten fotografi.

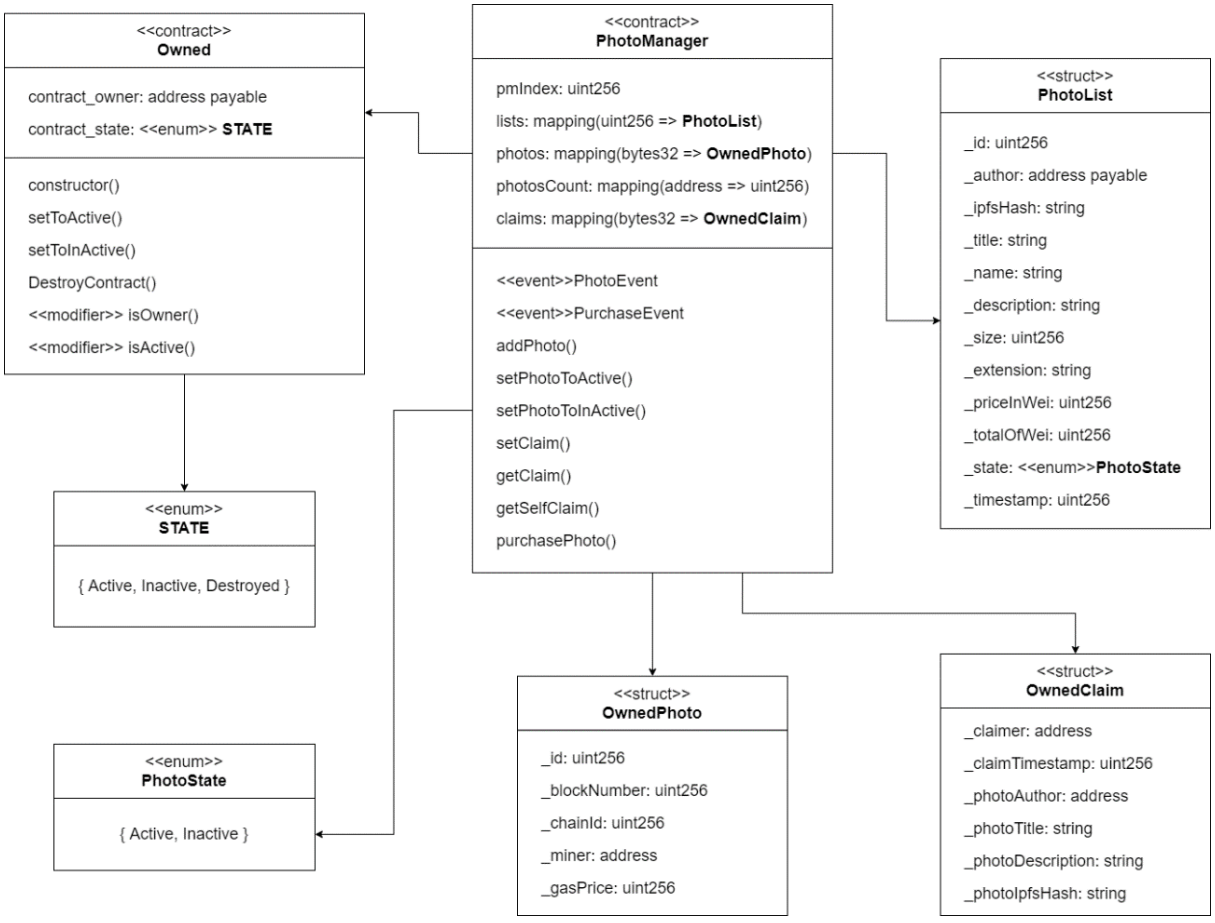
3.2 Arsitektur Sistem

Arsitektur sistem yang diterapkan pada sistem penyimpanan konten fotografi ditampilkan pada Gambar 4. Pertama smart contract akan compile dan di-deploy ke jaringan ethereum, pada proses ini smart contract akan diubah menjadi *Ethereum Virtual Machine Byte Code* (*EVM Byte Code*). Lalu web server yang memiliki file website akan menjadi menjadi *host* atau *seeder* pada IPFS, sehingga user dapat mengakses *website* dengan memberikan *hash ipfs* yang sudah ditentukan pada bagian alamat di browser. Ketika *website* dibuka maka web3js akan melakukan API call dengan metode *RPC Interface* pada *smart contract instance* untuk dapat berkomunikasi dengan ethereum blockchain agar luar *blockcain* (*client*) bisa melakukan akses *read* dan *write* ke jaringan *ethereum*. Ketika user melakukan transaksi untuk menyimpan konten fotografi, maka *InterPlanetary File System* akan dipanggil melalui *http-ipfs-client* lalu konten fotografi akan tersimpan ke *IPFS* dan *IPFS* akan melakukan *return hash*. *Hash* yang diberikan oleh *IPFS* yang pada akhirnya disimpan pada jaringan *Ethereum*. Saat transaksi sudah divalidasi atau di-mining oleh *miner*, maka konten fotografi akan muncul secara otomatis pada halaman *website* dengan proses yang melibatkan *web3js* dan *http-ipfs-client*.



Gambar 4. Arsitektur Sistem

3.3 Struktur Smart Contract



Gambar 5. Struktur Smart Contract

Berdasarkan struktur smart contract yang telah ditampilkan pada Gambar 5 terdapat *Photo Manager Contract* untuk menyimpan informasi pada sistem konten fotografi dan *Owned Contract* untuk menyimpan informasi ownership pada contract saat *owner* melakukan *deploy smart contract* pada jaringan *Ethereum.*

4. PENGUJIAN SISTEM

4.1 Cost pada Ethereum

Pengujian *cost* pada Ethereum dilakukan untuk mendapatkan jumlah *cost* yang dibutuhkan oleh sistem konten fotografi. Parameter yang dibutuhkan dalam penghitungan adalah gas yang dibutuhkan, harga gas, harga ether terhadap rupiah. Data gas price yang ditampilkan pada Tabel 2 didapatkan dari *website etherscan* yang diakses pada tanggal 6 Juni 2021. *Gas price* dibagi menjadi 3 jenis yaitu *high*, *average*, dan *low*. Masing-masing jenis memiliki range diantaranya *max*, *average*, dan *min*. Penyesuaian *gas price* dilakukan dengan menambahkan jenis *gas price* dengan *range* yang sama kemudian dibagi 3. Data harga *ether* terhadap rupiah didapatkan dari website *coinmarket* yang diakses pada tanggal 6 Juni 2021, dengan harga Rp 37.511.701 per *ether* seperti yang ditampilkan pada Tabel 3.

Tabel 2. Gas Price pada Etherscan

Range	Gas Price in Gwei		
	High	Medium	Low
Max	100	36	16
Average	15	12	10
Min	11	10	5
Rata - rata	42	19	10

Tabel 3. Ether Price terhadap Rupiah

Ether (ETH)	Rupiah (IDR)
1	Rp37.511.701

Tabel 4. List Foto untuk Pengujian

No	IPFS Hash	Price in ETH
#1	QmR95tG8zcHzKJrRvNjdUR16KtB3KnCeddZA6AtVSF7nZh	0.1
#2	QmNXV8UnG7ZAsijsUpsfwuBj4hE9md22eNmHrTzhSKfYR	0.1
#3	QmXn8r25e9Z7neo4aRTwBvU2QDzBHsyB1XDzphnGww9TYD	0.1
#4	QmVn99fHEzCcyCkW2APeGvVQmyJni4ERvf1DJCBvt4PeWZ	0.1
#5	Qmf73gTfmTFzHnYc4grfARtaBiDmjaJCFPVTggctwM3xnM	0.1

Tabel 5. Pengajuan Cost pada Fitur Add dengan High Gas Price

Photo	Used Gas	Gwei	ETH	IDR
#1	515045	42	0.021632	Rp843,119.29
#2	431354	42	0.018117	Rp706,118.64
#3	543734	42	0.022837	Rp890,082.66
#4	521130	42	0.021887	Rp853,080.32
#5	431306	42	0.018115	Rp706,040.07
Total	2442569		0.102588	Rp3,998,440.98
Rata-rata	488514	42	0.020518	Rp799,688.20

Tabel 6. Pengajuan Cost pada Fitur Add dengan Average Gas Price

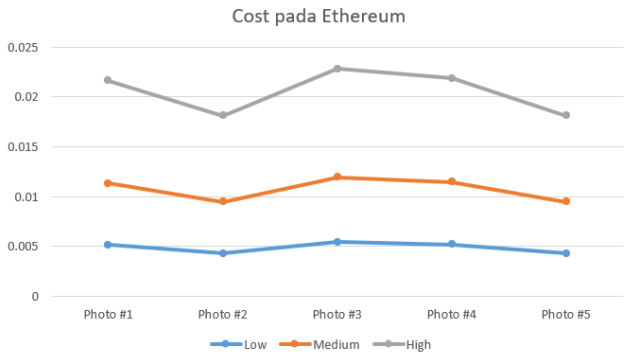
Photo	Used Gas	Gwei	ETH	IDR
#1	515045	19	0.011331	Rp441,633.91
#2	431354	19	0.00949	Rp369,871.67
#3	543734	19	0.011962	Rp466,233.77
#4	521130	19	0.011465	Rp446,851.60
#5	431306	19	0.009489	Rp369,830.51
Total	2442569		0.053737	Rp2,094,421.47
Rata-rata	488513.8	19	0.010747	Rp418,884.29

Tabel 7. Pengajuan Cost pada Fitur Add dengan High Gas Price

Photo	Used Gas	Gwei	ETH	IDR
#1	515045	10	0.00515	Rp200,742.69
#2	431354	10	0.004314	Rp168,123.49
#3	543734	10	0.005437	Rp211,924.44
#4	521130	10	0.005211	Rp203,114.36
#5	431306	10	0.004313	Rp168,104.78
Total	2442569		0.024426	Rp952,009.76
Rata-rata	488513.8	10	0.004885	Rp190,401.95

Gas yang diperlukan pada tiap foto berbeda karena data yang disimpan ke *storage variable* pada *ethereum blockchain* berbeda, *gas* yang diperlukan akan semakin meningkat jika data yang disimpan semakin banyak. Ukuran foto tidak secara signifikan mempengaruhi harga foto, karena yang disimpan pada *Ethereum blockchain* adalah *hash* yang didapatkan dari *IPFS* dengan ukuran 48 *string*. Data yang dikirim ke dalam *ethereum* berupa *string* dan int256 kemudian diformat menjadi *struct* dalam sebuah *mapping*.

Pada hasil pengujian Implementasi *Ethereum Claims Registry* dan *InterPlanetary File System* pada *High Gas Price* di Tabel 5, mendapatkan total sebanyak 0.102588 ETH atau Rp3,998,440.98 dan rata-rata sebanyak 0.020518 ETH atau Rp799,688.20. Pada *Average Gas Price* di Tabel 6 mendapatkan total sebanyak 0.053737 ETH atau Rp2,094,421.47 dan rata-rata sebanyak 0.010747 ETH atau Rp418,884.29. Pada *Low Gas Price* di Tabel 7 mendapatkan total sebanyak 0.024426 ETH atau Rp952,009.76 dan rata-rata sebanyak 0.004885 ETH atau Rp190,401.95. Pada Gambar 6 menampilkan perbandingan *cost ether* yang dibutuhkan dengan *gas price* yang telah disesuaikan.



Gambar 6. Grafik Cost pada Ethereum berdasarkan Gas Price

4.2 Perbandingan Ether dan Struktur pada ERC dengan Keccak256 dibandingkan ERC Orisinal

Perbandingan dilakukan melalui Remix – Ethereum IDE menggunakan *injected web3 environment* lalu *deploy smart contract* ke *Ropsten Test Network*. *Ether* yang digunakan untuk *deploy smart contract* merupakan *test ether* yang dapat di-*request* pada *Rospten Ethereum Faucet*. Transaksi dilakukan satu demi satu lalu lalu dicatat seperti yang ditampilkan pada Tabel 8, Tabel 9, dan Tabel 10.

Tabel 8. Perbandingan Ether pada *Deploy Smart Contract*

Deploy Smart Contract	Gas	G wei	ETH	IDR
ERC dan Keccak 256	2805011	10	0.0280501	Rp1,093,274 .27
ERC Orisinal	2833026	10	0.0283303	Rp1,104,193.33
Selisih	-2423 4		-0.0002802	-Rp10,946. 06
Efisiensi %	1%			

Tabel 9. Perbandingan Ether pada Transaksi *Make Claim*

Deploy Smart Contract	Gas	G wei	ETH	IDR
ERC dan Keccak 256	105844	10	0.0010584	Rp41,253.50
ERC Orisinal	105466	10	0.0010547	Rp41,106.17
Selisih	378		0.0000037	Rp147.33
Efisiensi %	-0.4%			

Tabel 10. Perbandingan Ether pada Transaksi Set Claim

Deploy Smart Contract	Gas	G wei	ETH	IDR
ERC dan Keccak 256	91696	10	0.000917	Rp35,739.21
ERC Orisinal	91306	10	0.000913	Rp35,587.20
Selisih	390		0.000004	Rp152.01
Efisiensi %	-0.4%			

Dari hasil pengujian diatas, maka *Ethereum Claims Registry* dan *Keccak256* dapat menghemat jumlah *ether* pada *deploy smart contract* dibandingkan *Ethereum Claims Registry* original, dengan selisih 24.234 gas dan 0.00028 ETH atau Rp10.919,06 seperti yang ditampilkan pada Tabel 8 Pada fitur *make claim Ethereum Claims Registry* dan *Keccak256* mempunyai beban *ether* lebih banyak, sebesar 378 gas dan 0.0000003 ETH atau Rp147,33 seperti yang ditampilkan pada Tabel 9 Pada fitur *set claim Ethereum Claims Registry* dan *keccak256* mempunyai beban *ether* lebih banyak, sebesar 390 gas dan 0.000004 ETH atau Rp152,01 yang ditampilkan pada Tabel 10.

Dengan implementasi *Ethereum Claims Registry* dan *Keccak256* mengubah struktur bentuk standar asal yang 3 dimensi menjadi satu dimensi, seperti yang ditampilkan pada Gambar 5.23 dan Gambar 5.24. Ethereum Claims Registry merupakan storage variable yang berbentuk mapping, memiliki key berupa bytes32 dan memiliki value berupa struct dari OwnedPhoto. Sedangkan Ethereum Claims Registry Orisinal merupakan storage variable yang berbentuk nested mapping dan value berupa struct Owned Photo, sehingga membentuk pola 3 dimensi. Ethereum Claims Registry dan Keccak256 lebih baik bila diimplementasikan pada struktur dengan dimensi lebih dari satu dengan perbedaan *struktur* hanya ada pada *storage variable* dan akan mempengaruhi cost pada smart contract. Transaksi berisikan status, transaction hash, from, to, transaction cost, hash, input, decoded input, logs, dan value. Namun pada untuk transaksi tidak ada perbedaan pada input, hanya perbedaan pada to, hash, dan logs karena mempunyai transaction hash yang berbeda. Smart contract hanya di-deploy satu kali oleh pemilik dari smart contract, ketika smart contract sudah divalidasi oleh miner maka smart contract dapat digunakan pada system. Setelah itu user dapat melakukan operasi make claim dan set claim pada sistem yang terintegrasi dengan jaringan *ethereum blockchain*. Pada Solidity versi pragma 8.3 masih dimungkinkan pemilik smart contract untuk melakukan self desctruct pada smart contract yang dibuat, sehingga smart contract tidak dapat digunakan mesti semua transaksi yang terdapat di jaringan ethereum tetap ada.

```
mapping(bytes32 => OwnedPhoto) public photos;

function setClaim(address _author, uint256 _id) public {
    require(lists[_id]._author == msg.sender, "You are not the author");

    PhotoList memory getPhoto = lists[_id];
    bytes32 encryptedBytes = keccak256(abi.encode(msg.sender, getPhoto._author, _id));

    claims[encryptedBytes] = OwnedClaim (
        msg.sender, block.timestamp,
        getPhoto._author,
        getPhoto._title,
        getPhoto._description,
        getPhoto._ipfsHash
    );
}
```

Gambar 7. Struktur pada Implementasi Ethereum Claims Registry dan Keccak256

```
mapping(address => mapping(address => mapping(uint => OwnedClaim))) private claims;

function setClaim(address _author, uint256 _id) public {
    require(lists[_id]._author == msg.sender, "You are not the author");

    PhotoList memory getPhoto = lists[_id];

    claims[msg.sender][getPhoto._author][_id] = OwnedClaim (
        msg.sender,
        block.timestamp,
        getPhoto._author,
        getPhoto._title,
        getPhoto._description,
        getPhoto._ipfsHash);
}
```

Gambar 8. Struktur pada Implementasi Ethereum Claims Registry Orisinal

Dari hasil pengujian perbandingan *cost* dan struktur, dari tabel dan gambar diatas maka dapat disimpulkan bahwa implementasi *Ethereum Claims Registry* dan *Keccak256* lebih hemat gas saat *deploy* namun untuk transaksi *make claim* dan *set claim* mendapatkan beban gas lebih banyak daripada *Ethereum Claims Registry* Orisinal. Struktur pada *Ethereum Claims Registry* dan *Keccak256* memakan gas lebih banyak karena terdapat proses enkripsi di dalam smart contract dibandingkan *Ethereum Claims Registry* Orisinal.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian sistem yang telah dilakukan dapat ditarik beberapa kesimpulan, antara lain :

- Berdasarkan hasil pengujian maka dapat disimpulkan bahwa implementasi metode *Ethereum Claims Registry* pada *Ethereum Blockchain* dan *InterPlanetary File System* dapat menghasilkan sebuah sistem konten fotografi yang dapat diakses secara global, menggunakan biaya yang relatif lebih murah dalam penyimpanan foto, dan memiliki fitur verifikasi atau claim.
- Website dan foto tersebar pada jaringan *InterPlanetary File System*, sehingga tidak terikat pada penyimpanan terpusat.
- *Ethereum Claims* Registry dapat digunakan untuk memberikan dan mengambil *claim* pada suatu konten fotografi.
- Transaksi *add photo* pada *high gas price* membutuhkan 0.020518 ETH, *average gas price* membutuhkan 0.010747, dan *low gas price* membutuhkan 0.004885. Jumlah eth dipengaruhi oleh banyaknya informasi yang dikirim dan diproses oleh *smart contract*. Semakin tinggi *gas price* maka biaya yang dibutuhkan semakin meningkat dan proses validasi transaksi oleh *miner* akan semakin cepat.
- Implementasi *Ethereum Claims Registry* dan *keccak256* dibandingkan *Ethereum Claims Registry* yang orisinal pada *deploy smart contract* dapat menghemat 24234 gas atau setara dengan 0.00028 ETH dan dengan tingkat efisiensi 1%, *make claim* lebih mahal 378 gas atau setara dengan 0.0000037 ETH dan dengan tingkat efisiensi -0.4%, dan *set claim* lebih mahal 390 gas atau setara dengan 0.000004 dan dengan tingkat efisiensi -0.4%.

5.2 Saran

- Melakukan implementasi metode *digital watermarking* untuk melakukan proteksi foto dan autentikasi untuk pengecekan orisinalitas dan kerusakan foto pada sistem konten fotografi.
- Proses enkripsi untuk claim yang menggunakan *keccak256* dilakukan diluar smart contract agar tidak menambah beban gas pada saat transaksi diproses oleh *miner*.
- Melakukan implementasi ERC-721 tentang *Non Fungible Token Standard* pada sistem konten fotografi agar transaksi dibatasi oleh pemilik yang memiliki token tertentu.
- Memperbaharui *user interface* pada sistem konten fotografi agar lebih menarik dan lebih *user friendly*.

6. DAFTAR PUSTAKA

[1] Alharby, M., & Moorsel, A. 2017. Blockchain Based Smart Contracts : A Systematic Mapping Study. *Computer Science & Information Technology (CS & IT)*, 3. https://doi.org/10.5121/csit.2017.71011

[2] Atlam, H. F., & Wills, G. B. 2019. Technical aspects of blockchain and IoT. *Advances in Computers*, 1–39. https://doi.org/10.1016/bs.adcom.2018.10.006

[3] Bhowmik, D., & Feng, T. 2017. The multimedia blockchain: A distributed and tamper-proof media transaction framework. *2017 22nd International Conference on Digital Signal Processing (DSP)*. Published. https://doi.org/10.1109/icdsp.2017.8096051

[4] Cox, S. 2020. Introduction to Photography: The Universal Language. *Photography Life*. URI= https://photographylife.com/what-is-photography

[5] Ekparinya, E., Gramoli, V., & Jourjon, G. 2018. Double-Spending Risk Quantification in Private, Consortium and Public Ethereum Blockchains. https://arxiv.org/pdf/1805.05004.pdf

[6] Express New. 2019. Coimbatore wedding photographers 'made to pay' by ransomware. *Newindianexpress*. URI=https://www.newindianexpress.com/states/tamil-nadu/2019/oct/12/coimbatore-wedding-photographers-made-to-pay-by-ransomware-2046273.html

[7] Kami, I. M. 2019. Tas Fotografer NatGeo Hilang di Bandara Bali, Foto Ratusan Spesies Ikut Raib. Detiknews. URI=https://news.detik.com/berita/d-4833216/tas-fotografer-natgeo-hilang-di-bandara-bali-foto-ratusan-spesies-ikut-raib

[8] Lesniscate. 2017. Photographer deleted our wedding photos, please help. Reddit. URI=https://www.reddit.com/r/WeddingPhotography/comments/745wvp/photographer_deleted_our_wedding_photos_please/

[9] Nyaletey, E., Parizi, R. M., Zhang, Q., & Choo, K.-K. R. 2019. BlockIPFS - Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability. *2019 IEEE International Conference on Blockchain (Blockchain)*, 19. https://doi.org/10.1109/blockchain.2019.00012

[10] Prashanth Joshi, A., Han, M., & Wang, Y. 2018. A survey on security and privacy issues of blockchain technology. *Mathematical Foundations of Computing*, *1*(2), 126. https://doi.org/10.3934/mfc.2018007

[11] Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. 2018. Blockchain-Based, Decentralized Access Control for IPFS. *2018 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 1500. https://doi.org/10.1109/cybermatics_2018.2018.00253