

IMPLEMENTASI BLOCKCHAIN UNTUK PENDATAAN DOKUMEN DIGITAL (IMPLEMENTATION OF BLOCKCHAIN FOR DIGITAL DOCUMENT DATA COLLECTION)

1st Timothy Harlian
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

tharlian@student.telkomuniversity.ac.id

2nd Yudha Purwanto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

omyudha@telkomuniversity.ac.id

line 1: 3rd Muhammad Faris Ruriawan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

muhammadfaris@telkomuniversity.ac.id

Abstrak

Pada saat ini banyak orang sudah mulai beralih dari dokumen secara fisik ke dokumen secara digital. Dokumen secara fisik mungkin hanya dibutuhkan ketika sedang mengurus hal-hal yang berbau administratif dan membutuhkan *copy* dari dokumen tersebut sebagai bukti fisik. Pada saat ini dokumen digital banyak diimplementasikan pada berbagai macam hal seperti sertifikat, piagam penghargaan, dan bahkan ijazah. Namun hal tersebut rawan akan pemalsuan dan jika hal tersebut dilakukan pendataan, akan sulit untuk dideteksi bila pada *database* ada yang mengubah data tersebut atau informasi mengenai data tersebut karena kurangnya transparansi. Maka dari itu, *blockchain* bisa menjadi salah satu cara atau jalan keluar untuk mengatasi hal tersebut. Karena *blockchain* unggul dalam hal transparansi data sehingga data tersebut akan terjaga validitasnya. Selain itu karena *blockchain* hanya menggunakan sistem *read and write* pada setiap transaksinya maka transaksi-transaksi yang sudah dilakukan akan tetap ada dan tidak berubah. Hasil yang didapat dari Tugas Akhir ini adalah, terciptanya aplikasi website berbasis *blockchain* menggunakan *blockchain Ethereum*, serta terkoneksi dengan jaringan *Rinkeby Test Network* sebagai jaringan *blockchain*. Aplikasi *blockchain* ini dapat melakukan *input, retrieve, update, dan delete* dengan keberhasilan *black box testing* sebesar 100% dan *white box testing* sebesar 100%. Pengujian performa untuk seluruh transaksi memiliki waktu paling lambat 6.04 detik dan paling cepat 93.32 detik.

Kata Kunci: *blockchain, basis data, transparansi*

Abstract

Nowadays lot of people switch from physical document to digital document. Maybe people only use physical document only for administrative matters and need the copy from the document as physical evidence. Digital document implemented in various ways like certificate, certificate of merit, even diploma. But those things vulnerable to counterfeit and if we do data collection, it will be hard to detect if someone do any changes about the information of document at database because of less of transparency. Therefore, blockchain can be the way out to overcome this problem. Because blockchain excellent in data transparency so the validity is maintained. Because blockchain only using

read and write for the transaction, then the transaction has been done will stay and cannot change. Henceforth, result obtained from this thesis is, creatin web application blockchain based use Ethereum blockchain also connected with Rinkeby Test Network as blockchain network. This blockchain application can do input, retrieve, update, and delete with 100% black box testing, and 100% white box testing. Performance testing for all transaction have slowest time 6.04 second and fastest time 93.32 second.

Keywords: blockchain, database, transparency

I. PENDAHULUAN

Seiring berjalannya perkembangan teknologi, dokumen-dokumen tentu saja terkena dampaknya. Yang awalnya dokumen dalam bentuk fisik, namun sekarang muncul dokumen yang berbentuk digital. Dan tentunya perkembangan dokumen digital ini memberikan manfaat contohnya seperti dokumen yang lebih mudah disimpan, mudah untuk di *backup*, dapat di transfer menggunakan *internet*, dapat meningkatkan keamanan, tidak khawatir oleh rayap, dan juga lebih ramah lingkungan[1].

Untuk memastikan dokumen-dokumen tersebut valid dan terdaftar, tentu saja membutuhkan sistem pendataan. Pendataan secara *online* dapat menggunakan aplikasi berbasis website dan juga menggunakan basis data sehingga pendataan menjadi lebih efisien. Namun permasalahannya adalah basis data tradisional tidak transparan. Serta sulitnya melakukan *tracking* apabila ada orang yang dengan sengaja maupun tidak sengaja merubah pendataan dari dokumen tersebut. Namun karena *blockchain* memiliki sistem yang hanya bersifat *read and write* pada setiap transaksinya, hal ini dapat mempermudah *tracking* apa bila ada yang berubah dari datanya.

Maka dari itu pada penelitian ini akan melakukan perancangan dan juga pembuatan aplikasi yang dapat melakukan pendataan dokumen digital. Aplikasi yang dirancang akan menggunakan sistem *blockchain*. Diharapkan dengan adanya aplikasi pendataan yang menggunakan *blockchain* dapat menjadi jawaban dari permasalahan yang ada.

Dari Tugas Akhir ini identifikasi masalah yang didapatkan antara lain sistem *blockchain* seperti apa yang sesuai untuk melakukan pendataan dokumen digital, dan bagaimana implementasi *blockchain* sebagai pendataan dokumen digital. Tujuan dan manfaat dari Tugas Akhir ini adalah, mengetahui cara pembuatan website berbasis *blockchain Ethereum* dan mengetahui cara mengimplementasikan *blockchain Ethereum* pada website pendataan dokumen digital. Dalam pengerjaan Tugas Akhir ini

menggunakan metode penelitian seperti studi literatur, perancangan sistem, pengujian alat, serta penyusunan buku tugas akhir.

II. KAJIAN TEORI

A. Blockchain

Blockchain pertama kali diperkenalkan oleh Satoshi Nakamoto dalam pengembangannya untuk *bitcoin*. Hal ini didasari karena kasus kejatuhan *Lehman Brothers*, yang membuat Satoshi Nakamoto memikirkan bagaimana mengirimkan uang secara langsung tanpa melewati perantara[2]. *Blockchain* sendiri merupakan sebuah buku besar yang mencatat setiap transaksi dan juga mengamankan transaksi tersebut dan tidak menggunakan pihak ketiga untuk melakukan pertukaran. Dimana contohnya seperti ketika kita akan melakukan transaksi di sebuah tempat, jika kita menggunakan sistem debit ataupun credit pihak ketiganya adalah bank yang kita gunakan. Bagaimana jika pihak ketiga mengalami masalah atau gangguan? Tentu saja transaksi pembayaran akan terhambat. Selain itu dengan adanya *blockchain* yang memiliki sistem terdesentralisasi, semua catatan transaksi akan dapat dilihat secara langsung[3]

B. Ethereum

Pada dasarnya *Ethereum* sendiri adalah sebuah sistem *blockchain* yang sama dengan yang lainnya namun *Ethereum* mempunyai mata uang kripto sendiri yang disebut dengan *ether* (ETH). Konsep daripada ETH sendiri kurang lebih sama seperti *Bitcoin*, namun *Ethereum* dapat digunakan untuk membuat aplikasi lain yang dimana aplikasi ini dinamakan dengan aplikasi terdesentralisasi[4].

C. Smart Contract

Dalam pembuatan *blockchain* tidak terlepas dari yang namanya *smart contract*. Awal mulanya *smart contract* dicetuskan oleh ahli computer yang merupakan sarjana hukum dan juga pakar kriptografi bernama Nick Szabo. Dengan adanya *smart contract* ini lah yang membuat transaksi digital menjadi transparan dan juga hanya bisa diakses dengan orang yang berada di jaringan yang sama[5]. Untuk sistem *Ethereum* sendiri *smart contract* yang digunakan adalah bahasa pemrograman *Solidity*. Berbeda dengan bahasa pemrograman lainnya, jika kita mengenal PBO atau Pemrograman Berorientasi Objek maka untuk *Solidity* ini sendiri merupakan Pemrograman Berorientasi Kontrak[6].

D. Metamask

Metamask merupakan sebuah dompet *cryptocurrency* khususnya Ethereum, dan token Ethereum dengan protokol ERC-20 dan ERC-721. *Metamask* didirikan Aaron Davis dan Consesys. *Metamask* adalah ekstensi *browser* dan dapat digunakan pada Chrome, Firefox, Brave, dan Microsoft Edge[7]. Selain sebagai dompet *cryptocurrency*, *metamask* berguna untuk menghubungkan pengguna dengan aplikasi desentralisasi Ethereum. Dengan adanya *metamask*, membuat proses transaksi pada aplikasi terdesentralisasi menjadi lebih aman. Sebelumnya aplikasi terdesentralisasi harus memberikan *private key* ke *cryptocurrency wallet*, namun dengan adanya *metamask* aplikasi terdesentralisasi

cukup meminta izin untuk melakukan transaksi dana melalui *metamask*[8].

E. Web3.js

Web3.js merupakan sebuah JavaScript *library* berbasis Ethereum untuk berinteraksi ke *node* Ethereum baik secara lokal maupun *remote*. Untuk menggunakan *library* ini, pastikan sistem sudah memiliki Node.js dan NPM [9]. *Web3.js* berkomunikasi dengan *blockchain* Ethereum menggunakan sebuah protokol bernama "*Remote Procedure Call*" atau yang biasa dikenal dengan JSON RPC. Protokol tersebut memungkinkan klien untuk membaca ataupun menulis ke jaringan *blockchain*, dan *Web3.js* memungkinkan untuk mengirim permintaan ke *node* Ethereum[10].

III. HASIL DAN PEMBAHASAN

A. Pengujian White Box

Pada pengujian *white box* dilakukan beberapa skenario yang berbeda yaitu skenario ketika melakukan *input* informasi dokumen, *input* informasi verifikasi, mencari informasi dokumen, mencari informasi verifikasi, melakukan *update* informasi dokumen dan verifikasi, serta menghapus informasi dokumen dan verifikasi. Pada pengujian skenario pertama, aplikasi berjalan sesuai ekspektasi, yaitu pengguna dapat melakukan *input* informasi dokumen dari awal hingga akhir. Pengguna juga dapat melakukan *input* verifikasi sesuai dengan *form* yang disediakan hingga tercatat pada sistem *blockchain*. Skenario berikutnya adalah ketika pengguna melakukan *update* informasi mengenai dokumen dan verifikasi. Pada skenario ini, aplikasi berjalan sesuai ekspektasi yang mana pengguna dapat mengisi *form update* sesuai yang dibutuhkan dan akhirnya tercatat pada sistem *blockchain*. Selanjutnya adalah skenario mencari dokumen dan verifikasi, pengguna dapat mencari dokumen dan juga verifikasi yang sudah dimasukkan ataupun diubah dengan melakukan *input* nomor dokumen. Skenario ini berjalan sesuai dengan ekspektasi karena aplikasi dapat mencari informasi dokumen maupun verifikasi yang sudah dimasukkan atau diubah. Terakhir adalah skenario untuk menghapus informasi dokumen maupun verifikasi, skenario ini pun berjalan sesuai ekspektasi karena aplikasi dapat menghapus informasi dokumen ataupun verifikasi.

B. Pengujian Black Box

Pengujian *black box* dilakukan untuk memastikan bahwa setiap fitur yang ada pada aplikasi berjalan dengan baik dan sesuai ekspektasi. Karena pada aplikasi ini memiliki skenario untuk melakukan *input* dokumen dan verifikasi, melihat informasi dokumen dan verifikasi, *update* dokumen dan verifikasi, serta menghapus dokumen dan verifikasi setiap skenario pasti memiliki fitur-fitur yang disesuaikan. Pada *input* dokumen memiliki *form* dengan *field input* nomor dokumen, nama dokumen, nama pembuat dokumen, unit pembuat dokumen, NIP pembuat dokumen, nama penanda tangan dokumen, tanggal tanda tangan dokumen, serta tombol untuk melakukan *input*. Semua fitur yang ada berjalan sesuai ekspektasi karena pengguna dapat memasukkan

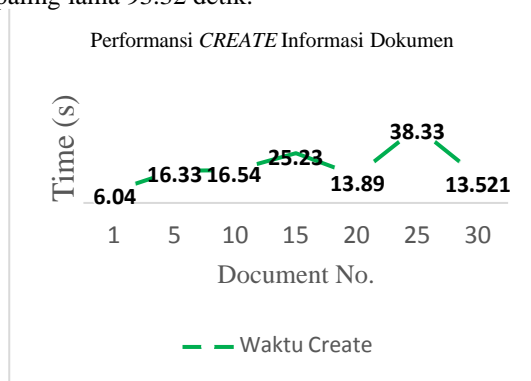
informasi dokumen dan menyimpannya ke sistem *blockchain*.

Fitur-fitur untuk *input* verifikasi seperti *input field* nomor dokumen, nama dokumen, nama verifikasi, unit verifikasi, NIP verifikasi, tanggal verifikasi, serta tombol untuk melakukan *input* verifikasi berjalan sesuai dengan ekspektasi karena pengguna dapat melakukan *input* informasi verifikasi ke sistem. Selanjutnya ada *edit* dokumen, fitur yang diberikan kurang lebih sama seperti *input* dokumen seperti *input field*. Namun tombol yang diberikan berbeda yaitu tombol untuk *edit* dokumen. Semua fitur yang ada sesuai ekspektasi karena pengguna dapat merubah informasi dokumen. Fitur berikutnya adalah *edit* verifikasi, fitur *form* yang diberikan sama seperti *input* dokumen namun yang membedakan adalah tombol *edit* verifikasi. Semua berjalan sesuai dengan ekspektasi karena informasi verifikasi dapat dirubah.

Berikutnya ada fitur untuk menghapus dokumen dan verifikasi, keduanya sama-sama diberikan *input field* nomor dokumen. Namun yang dibedakan adalah tombol untuk menghapus dokumen dan verifikasi serta informasi yang didapatkan. Semua elemen berjalan sesuai ekspektasi, *input field* dapat dimasukkan nomor dokumen, serta tombol hapus dokumen dan hapus verifikasi dapat menjalankan fungsinya. Ketika hapus dokumen berhasil maka akan menampilkan informasi bahwa dokumen tersebut telah dihapus, begitu pun jika gagal. Sama seperti hapus dokumen, ketika kita melakukan hapus verifikasi dan berhasil maka akan menampilkan bahwa verifikasi berhasil dihapus, begitu pun sebaliknya.

C. Pengujian Performansi

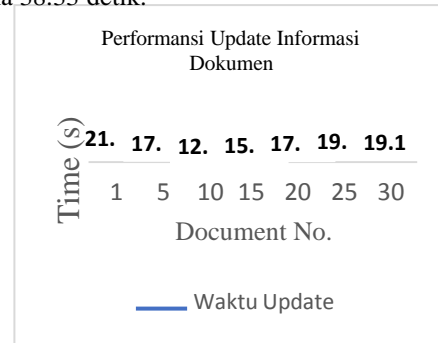
Uji performansi dilakukan pada fitur *input*, *update*, dan *delete*. Hasil dari uji performansi ini, berupa berapa lama waktu yang dibutuhkan untuk melakukan setiap transaksi ke sistem *blockchain*. Satuan yang digunakan dalam uji performansi ini adalah satuan detik. Dari semua hasil uji performansi didapatkan bahwa *blockchain* memiliki waktu yang berbeda-beda dalam melakukan sebuah transaksi. Dari keseluruhan transaksi, waktu paling cepat adalah 6.04 detik dan paling lama 93.32 detik.



Gambar 1. Grafik Performansi *Create* Informasi Dokumen

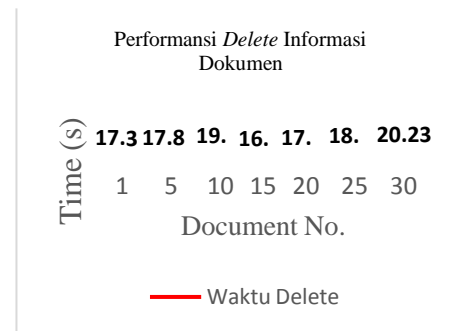
Berdasarkan Gambar 1, ditunjukkan performa dari *input* data. Grafik tersebut menunjukkan berapa lama waktu yang dihasilkan untuk melakukan *input* informasi dokumen. Pada gambar di atas terlihat bahwa performansi untuk melakukan

input dokumen dibutuhkan waktu paling cepat 6.04 detik, dan paling lama 38.33 detik.



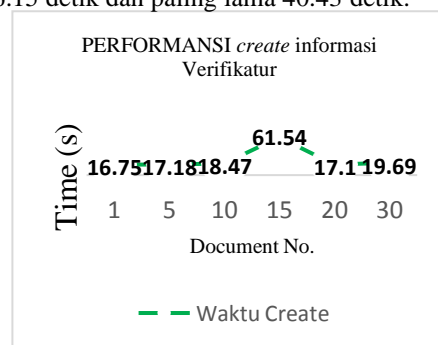
Gambar 2. Grafik Performansi *Update* Informasi Dokumen

Berdasarkan Gambar 2, menunjukkan performa ketika melakukan *update* data. Grafik tersebut menunjukkan waktu yang dihasilkan untuk melakukan *update* informasi dokumen. Hasil performansi untuk melakukan *update* dari informasi dokumen yang sudah dimasukkan membutuhkan waktu paling cepat 8.43 detik dan paling lama 38.08 detik.



Gambar 3. Grafik Performansi *Delete* Informasi Dokumen

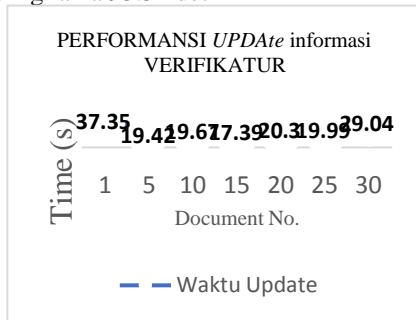
Gambar 3 menunjukkan grafik performansi ketika melakukan penghapusan data dari dokumen. Grafik tersebut menunjukkan waktu yang dihasilkan oleh sistem *blockchain* ketika melakukan hapus data. Dalam kasus ini, waktu tercepat 16.15 detik dan paling lama 40.43 detik.



Gambar 4. Grafik Performansi *Create* Informasi Verifikator

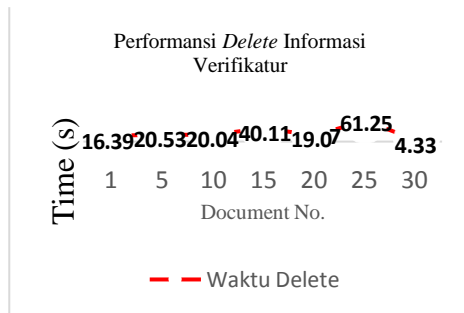
Selain informasi dokumen, sistem ini juga mencatat siapa yang melakukan verifikasi pada dokumen tersebut. Gambar 4 menunjukkan hasil performansi waktu yang dihasilkan ketika melakukan *input* informasi verifikasi. Pada pencatatan data

verifikatur membutuhkan waktu paling cepat sekitar 10.01 detik dan paling lama 93.32 detik



Gambar 5. Grafik Performansi Update Informasi Verifikatur

Gambar 5 menunjukkan hasil uji performansi ketika melakukan *update* untuk informasi verifikatur. Untuk melakukan *update* atau perubahan informasi verifikatur. Dibutuhkan waktu paling cepat 15.3 detik dan paling lama 37.71 detik.



Gambar 6. Grafik Performansi Delete Informasi Verifikatur

Serta yang terakhir untuk menghapus informasi verifikatur. Gambar 6 menunjukkan berapa lama waktu yang dibutuhkan untuk menghapus informasi verifikatur. Dalam menghapus informasi verifikatur membutuhkan waktu paling cepat 4.33 detik dan paling lama 61.25 detik.

IV. KESIMPULAN

Berdasarkan seluruh pengujian aplikasi ini, seluruh skenario dan fitur-fitur yang ada pada aplikasi dapat berjalan sesuai ekspektasi. Baik pengujian *white box*, *black box*, serta uji performansi memberikan *output* yang sesuai dengan ekspektasi penulis. Tidak ada *bug* yang membuat aplikasi ini berjalan tidak sesuai dengan ekspektasi. Bahkan *tools* penunjang pun dapat berjalan sesuai skenario yang diharapkan. Berdasarkan seluruh pengujian aplikasi ini, seluruh skenario dan fitur-fitur yang ada pada aplikasi dapat berjalan sesuai ekspektasi. Berikut adalah hasil analisis dari setiap pengujian Berdasarkan *white box testing* aplikasi dapat berjalan sesuai dengan skenario algoritma yang sudah ditentukan. Seperti skenario untuk *input*, *edit*, *read*, *delete* informasi dokumen dan juga skenario *input*, *edit*, *read*, *delete* informasi verifikatur. Berdasarkan *black box testing* fitur yang ada di *website* dapat berjalan 100% sesuai dengan skenario. Setiap *input field* yang ada pada *website* dapat digunakan untuk mengisi informasi yang dibutuhkan, dan setiap tombol yang ada pada aplikasi dapat berjalan untuk

memanggil setiap *event*. Uji performa untuk fitur *create* dokumen memiliki waktu rata-rata senilai 20.21336667 detik, *update* dokumen memiliki waktu rata-rata senilai 19.31533 detik, *delete* dokumen memiliki waktu rata-rata 20.5581 detik Uji performa untuk fitur *create* verifikatur memiliki waktu rata-rata senilai 23.09833 detik, *update* verifikatur memiliki waktu rata-rata senilai 20.978 detik, dan *delete* verifikatur memiliki waktu rata-rata senilai 20.93067 detik.

REFERENSI

- [1] M. Digitalsense, "Untuk Apa sih Dokumen Digital?," <http://digitalsense.co.id/>, 2015. <http://digitalsense.co.id/tips-news/untuk-apa-sih-dokumen-digital> (accessed Nov. 19, 2020).
- [2] Rheinm, "Blockchain Adalah," <https://digitalis.id/>, 2020. <https://digitalis.id/blog/blockchain-adalah/> (accessed Nov. 19, 2020).
- [3] T. ID, "Apa sih Blockchain itu? Yuk Mengenal Definisi Blockchain dari sekarang!," <https://www.techfor.id/>, 2019. <https://www.techfor.id/apa-sih-blockchain-itu-yuk-mengenal-definisi-blockchain-dari-sekarang/> (accessed Nov. 19, 2020).
- [4] ethereum.org, "Apa Itu Ethereum?," <https://ethereum.org/>, 2020. <https://ethereum.org/id/what-is-ethereum/> (accessed Nov. 19, 2020).
- [5] Iqbalsweden, "Mengenal Lebih Detail Mekanisme Smart Contract di Blockchain," <https://steemit.com/>, 2017. <https://steemit.com/blockchain/@iqbalsweden/mengenal-lebih-detail-mekanisme-smart-contract-di-blockchain> (accessed Nov. 19, 2020).
- [6] D. Priyandani, "Apa itu Solidity Blockchain tle," <https://cryptoiz.net/>, 2020. <https://cryptoiz.net/apa-itu-solidity-blockchain/> (accessed Nov. 19, 2020).
- [7] W. Vermaak, "What Is MetaMask?," <https://coinmarketcap.com/>, 2021. <https://coinmarketcap.com/alexandria/article/what-is-metamask#toc-metamask-a-brief-history-and-key-features> (accessed Jul. 31, 2021).
- [8] J. Hendy, "Metamask wallet review and guide," <https://www.finder.com/>, 2021. <https://www.finder.com/metamask-review> (accessed Jul. 31, 2021).
- [9] S. Shukla, "Interacting With Ethereum Smart Contract Using Web3js," <https://www.geeksforgeeks.org/>, 2020. <https://www.geeksforgeeks.org/interacting-with-ethereum-smart-contract-using-web3js/>.
- [10] G. McCubbin, "Intro to Web3.js · Ethereum Blockchain Developer Crash Course," <https://www.dappuniversity.com/>, 2021. <https://www.dappuniversity.com/articles/web3-js-intro>.