IMPLEMENTASI ELECTRONIC SIGNATURE PADA JARINGAN PRIVAT BLOCKCHAIN ETHEREUM

Implementation of Electronic Signature on the Ethereum Blockchain Private Network

Bilya Putra Aji[1], Heri Wijayanto[1], Ahmad Zafrullah Mardiansyah[1,2]
[1]Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: bilyaputra@gmail.com, : [heri, zaf]@unram.ac.id

Abstract **Electronic signatures have advantages such as convenience and cost savings. However, electronic signatures have security weaknesses, such as the inability to be validated and maintain data integrity. To overcome these issues, blockchain technology is used, which has a high level of security and features a digital signature that functions to validate the signature. Therefore, an electronic signature system will be created using the private Ethereum blockchain network. The system is tested using unit testing of smart contracts, blackbox testing, accessing the system on the local network, and analyzing the three basic properties of blockchain: immutability, transparency, and decentralization. The results of the smart contract unit testing show that each function in the smart contract has operated as expected, and the blackbox testing results indicate that the system functions correctly. Additionally, the system can also be accessed by other computers as long as they are on the same network as the server computer.**

Key words**: *Blockchain, Electronic Signature, Decentralized Application, Ethereum, Website***

I. PENDAHULUAN

Tanda tangan merupakan identitas diri seseorang dengan menggunakan tanda tulisan sebagai keabsahan/sah tidaknya sebuah dokumen. Tanda tangan umumnya dibuat menggunakan pena yang biasa disebut tanda tangan basah. Di era digital ini, terdapat tanda tangan elektronik (¬e-sign) yang fungsinya sama dengan tanda tangan yang ditulis dengan tangan namun didigitalisasi. Akan tetapi, tanda tangan elektronik memiliki kelebihan seperti efisiensi waktu dan hemat biaya pengeluaran dikarenakan hanya membutuhkan koneksi internet dan perangkat keras seperti komputer dan telepon pintar (smartpohone), sehingga tanda tangan elektronik terlihat lebih unggul dari pada tanda tangan basah [1].

Tanda tangan elektronik dapat membantu dalam mempercepat proses administrasi seperti pada perguruan tinggi baik mahasiswa maupun dosen, dikarenakan tanda tangan elektronik cukup sering digunakan untuk berbagai keperluan seperti KRS dan laporan. Dibalik semua kelebihan tanda tangan elektronik tersebut, terdapat kelemahannya yaitu tidak memiliki keamanan sehingga suatu tanda tangan tidak dapat divalidasi dan juga tidak dapat menjamin integritas suatu dokumen [2]. Namun masalah tersebut dapat diselesaikan dengan membuat sebuah website yang berfungsi untuk membuat tanda tangan elektronik yang dilengkapi dengan kode QR sehingga tanda tangan tersebut dapat divalidasi. Akan tetapi, pengimplementasiannya masih menggunakan konsep sistem sentralisasi dalam penyimpananya, di mana pada sistem ini terdapat kelemahan dalam keamanan data dikarenakan apabila seseorang memiliki akses ke database tersebut maka kerusakan data dapat terjadi dan sumber daya yang digunakan untuk melakukan hacking ke database sentral tidak terlalu besar [3].

Salah satu teknologi yang dapat membantu dalam menyelesaikan permasalahan mengenai keamanan data dan transaksi adalah blockchain. Blockchain atau dapat disebut juga buku besar terdistribusi merupakan suatu basis data atau penyimpanan terdistribusi yang bersifat transparan dan tersebar ke seluruh node pada sebuah jaringan sehingga kebenaran suatu data dapat dipastikan. Sistem ini dapat meningkatkan keamanan dan mengurangi risiko data yang korup. Blockchain merupakan solusi yang tepat dalam melakukan penyimpanan data berupa transaksi tanda tangan elektronik agar terjamin keamanannya. Selain itu, tedapat fitur digital signature yang nantinya dapat digunakan untuk memvalidasi suatu tanda tangan sehingga keaslian dari suatu tanda tangan dapat dijamin.

Pada penelitian ini, sistem blockchain yang akan diterapkan menggunakan sistem blockchain milik Ethereum. Ethereum sendiri merupakan sebuah platform open source yang menyediakan fungsi-fungsi yang lengkap dalam pembuatan jaringan blockchain. Pada jaringan blockchain, penyimpanan data dilakukan menggunakan smart contract. Smart contract merupakan sebuah program yang berisi aturan-aturan yang digunakan untuk mengatur perilaku sebuah jaringan blockchain, sehingga dapat digunakan untuk mengatur penyimpanan dan proses lainnya. Bahasa pemrograman yang digunakan dalam penulisan smart contract yaitu Solidity. Penyebaran smart contract ke jaringan blockchain dapat menggunakan Truffle. Truffle merupakan sebuah framework yang

menyediakan berbagai fungsi seperti kompilasi, penyebaran dan pengujian smart contract [4].

Dengan menggunakan teknologi blockchain, keamanan sebuah data dapat dijamin dikarenakan setiap transaksi akan dienkripsi menggunakan secure hash algorithm, kemudian transaksi tersebut dihubungkan dengan transaksi yang lainnya sehingga membentuk blok yang terhubung dan nantinya akan disebarkan ke seluruh node pada jaringan blockchain, sehingga transaksi tersebut dapat divalidasi oleh semua node dan tidak diperlukan sebuah server sentral sebagai penyimpanan data [5]. Pada penelitian ini, penulis akan mengimplementasikan tanda tangan elektronik pada jaringan privat blockchain Ethereum.

II. TINJAUAN PUSTAKA

Penelitian terkait mengenai pengimpelementasian teknologi blockchain pernah dilakukan sebelumnya. Penelitian tersebut membahas tentang media penyimpanan sertifikat digital pada website akademik menggunakan teknologi blockchain. Keamanan data dapat dijamin dikarenakan sifat dari blockchain itu sendiri yaitu tidak dapat diubah (immutable) [3]. Waktu yang diperlukan untuk memproses 200 transaksi kurang lebih 8 detik. Ruang penyimpanan yang diperlukan ketika jumlah blok menyentuh angka 1 juta yaitu 2,2 Gigabytes yang dirasa masih relatif kecil untuk teknologi saat ini. Sedangkan jika jumlah blok mencapai 10 juta diperlukan kapasitas penyimpanan sebesar 22,6 Gigabytes [3]. Sedangkan daya yang digunakan untuk melakukan mining pada jaringan Bitcoin adalah 100-500 Megawatt pertahunnya [6].

Penelitian serupa juga pernah dilakukan sebelumnya mengenai implementasi blockchain untuk pendataan dokumen digital untuk memastikan dokumen-dokumen tersebut valid dan terdaftar. Pada penelitian tersebut terdapat beberapa fitur yang dibuat yaitu input data, update, dan delete. Waktu yang diperlukan untuk meng-input data yaitu palingcepat adalah 6,04 detik dan paling lama 93,32 detik. Sedangkan waktu yang diperlukan untuk melakukan update data yaitu paling cepat 8,43 dan paling lama 38,08 detik. Terkahir, waktu yang diperlukan untuk menghapus data paling cepat yaitu 16.15 detik dan paling lama 40,43 detik [7].

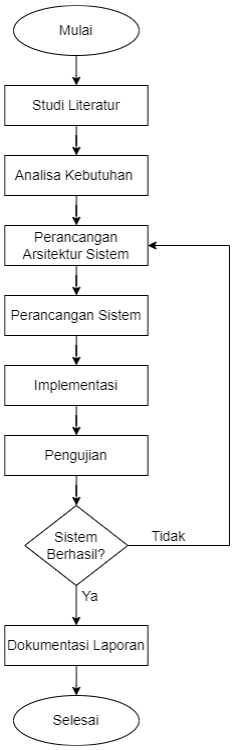
Penelitian berikutnya mengenai implementasi teknologi blockchain pada sistem pencatatan pemungutan suara. Pada penelitian tersebut, sistem yang telah dibuat memungkinkan proses pembuatan blok, verifikasi blok, dan juga penghitungan hasil voting. Penggunaan blockchain dapat mencegah terjadinya perubahan data voting dan karena sistem yang telah dibuat bersifat digital, proses penghitungan suara dapat dilihat secara real time. Sedangkan waktu yang diperlukan untuk memverfikasi 5.000 blok yaitu 10 detik [8]. Namun jika dibandingkan dengan database relasional, blockchain memiliki kecepatan membaca dan menulis lebih lambat [9].

Berdasarkan penelitian terkait yang telah dijelaskan di atas, penyimpanan data menggunakan teknologi blockchain terbukti aman, dikarenakan dari sifat blockchain itu sendiri yaitu terdesentraslisasi dan tidak dapat diubah. Sehingga apabila seseorang ingin mengubah suatu data pada jaringan blockchain memerlukan sumberdaya yang besar. Oleh karena itu, peneliti memutuskan untuk menggunakan teknologi blockchain sebagai media penyimpanan agar data transaksi tanda tangan elektronik menjadi lebih aman.

III. METODE PENELITIAN

A. Rencana Pelaksanaan

Adapun rencana pelaksanaan penelitian pengimplementasian tanda tangan elektronik pada jaringan privat *blockchain* dapat dilihat pada diagram alir berikut:



Gambar 1. Rencana Pelaksanaan

Berikut penjelasan dari diagram alir pada Gambar 3.1:

1. Pada tahap studi literatur dilakukan pengumpulan jurnal nasional maupun internasional yang berkaitan dengan perancangan sistem serupa dan telah ada sebelumnya, dalam hal ini pengimplementasian *blockchain*.
2. Tahap analisa kebutuhan sistem dilakukan analisis terhadap apa saja yang dibutuhkan untuk membangun sistem yang dirancang, menjelaskan perangkat yang dibutuhkan dalam proses perancangan serta pembangunan sistem.
3. Pada tahap perancangan arsitektur dilakukan perancangan keseluruhan arsitektur sistem dan alur kerja sistem yang dirancang.
4. Tahap perancangan sistem dilakukan perancangan web sederhana berdasarkan kebutuhan untuk bisa mengimplementasikan teknologi blockchain menggunakan smart contract pada e-sign.

5. Pada tahap implementasi merupakan tahapan dalam membuat dan menyusun perangkat lunak dengan Bahasa pemrograman. Bahasa pemrograman yang digunakan adalah Solidity dengan menggunakan Ganache sebagai blockchain lokal sekaligus database terdesentralisasi.
6. Tahap pengujian dilakukan dengan menggunakan *unit testing smart contract, blackbox testing*, mengakses sistem dengan komputer lain pada jaringan lokal dan analisis berdasarkan aspek dasar blockchain yaitu *immutable* atau tidak dapat diubah, transparan yaitu seluruh *node* dapat melihat semua transaksi yang ada pada *block*, dan terdesentralisasi yaitu setiap *node* menyimpan data yang sama.

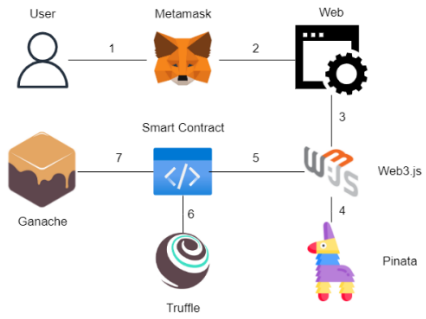
B. Analisa Kebutuhan

Pada tahap analisis kebutuhan sistem, persyaratan pengembangan akan dianalisis. Analisis yang dilakukan meliputi analisis kebutuhan alat dan bahan. Adapun alat dan bahan berupa perangkat keras dan perangkat lunak yang digunakan sebagai berikut:

1. Laptop dengan spesifikasi: Processor Intel® Core™ i5-8265U CPU @ 1.60GHz (8 CPUs), ~1.8GHz, RAM 12 GB dan Sistem Operasi Windows 10.
2. Aplikasi Ganache
3. Metamask

Aplikasi pendukung yang digunakan dalam pengembangan sistem yaitu text editor Visual Studio Code sebagai aplikasi untuk membangun web dengan menggunakan HTML dan CSS (Bootstrap), serta Web3.js sebagai penghubung antara smart contract blockchain dengan web.

C. Rancangan Arsitektur Sistem



Gambar 2. Rancangan Arsitektur Sistem

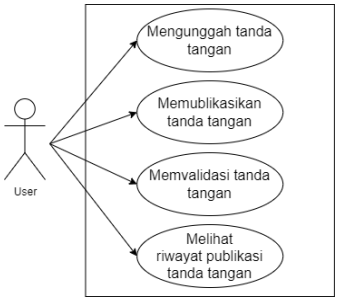
Pada tahap perancangan arsitektur sistem akan dilakukan perancangan terhadap arsitektur dan sistem yang akan dibangun. Berikut merupakan penjelasan dari rancangan arsitektur sistem di atas:

1. Pengguna diharuskan untuk *login* terlebih dahulu ke akun Metamask agar dapat berinteraksi dengan sistem.
2. Pada web, pengguna dapat melakukan *unggah* tanda tangan dan melakukan *digital signature* pada data menggunakan *private key* yang ada pada akun Metamask.
3. Berkas gambar yang diungah akan disimpan ke Pinata IPFS dan data tanda tangan yang telah di *input* oleh pengguna akan dikirim ke *smart contract* melalui web3.js.
4. Pinata IPFS berfungsi untuk menyimpan berkas tanda tangan pengguna.
5. Web3.js kemudian memanggil fungsi-fungsi yang ada pada *smart contract* untuk menyimpan data tanda tangan.
6. Di sini truffle berfungsi untuk membangun, melakukan *testing*, dan menyebarkan *smart contract* ke jaringan lokal *blockchain.*
7. Melalui *smart contract*, data tanda tangan tersebut akan disimpan ke jaringan lokal *blockchain* yaitu Ganache. Kemudian pengguna akan mendapatkan tanda tangan beserta kode QR dan *hash digital signature* yang nantinya dapat digunakan untuk memvalidasi tanda tangan tersebut pada web yang dibuat.

D. Perancangan Sistem

Berikut merupakan rancangan sistem yang akan dibuat:

D.1. Use Case Diagram



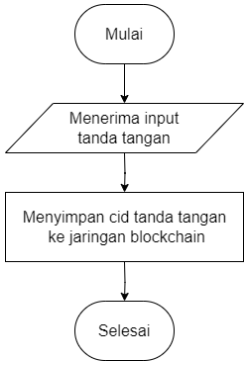
Gambar 3. *Use Case Diagram*

Pada gambar di atas merupakan rancangan use case diagram pada sistem yang akan dibuat. Berdasarkan use case tersebut, pengguna dapat mengunggah tanda tangan, memublikasikan tanda tangan ke jaringan blockchain, memvalidasi tanda tangan, dan melihat riwayat publikasi tanda tangan.

D.2. Flowchart Smart Contract

Berikut merupakan *flowchart smart contract* yang merupakan gambaran dari suatu proses dari sisi *smart contract.*

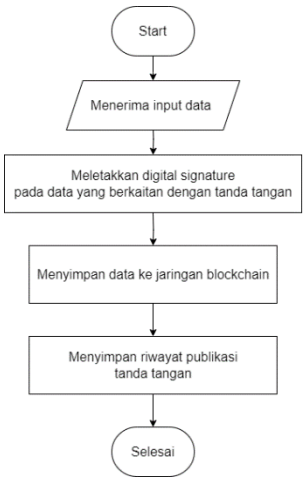
1. Mengunggah Tanda Tangan



Gambar 4. *Flowchart* Mengunggah Tanda Tangan

Berdasarkan flowchart pada gambar di atas, proses dimulai dengan smart contract menerima input berupa cid tanda tangan. Setelah itu, smart contract kemudian menyimpan cid tanda tangan tersebut ke dalam jaringan blockchain.

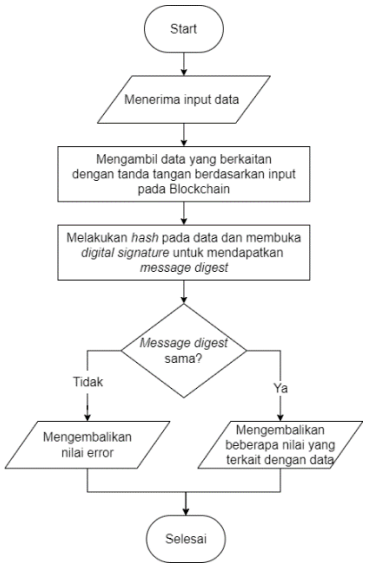
2. Memublikasi Tanda Tangan



Gambar 5. *Flowchart* Mengunggah Tanda Tangan

Berdasarkan *flowchart* pada gambar di atas, proses dimulai dengan *smart contract* menerima *input* data berupa *digital signature*, perihal, *cid* tanda tangan, waktu publikasi, akun penanda tangan, dan *cid* tanda tangan beserta kode QR. Lalu, akan diletakkan atau dipasangkan *digital signature* dengan data-data tersebut. Setelah itu, *smart contract* akan langsung menyimpan data tersebut ke dalam jaringan *blockchain*. *Smart contract* juga akan menyimpan riwayat publikasi tanda tangan berupa pemilik tanda tangan atau akun Metamask pengguna, waktu publikasi, *digital signature*, dan perihal.

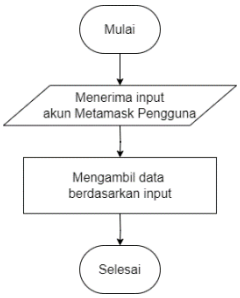
3. Memvalidasi Tanda Tangan



Gambar 6. *Flowchart* Memvalidasi Tanda Tangan

Berdasarkan *flowchart* pada gambar di atas, proses dimulai dengan *smart contract* menerima data berupa kode unik *digital signature* dari tanda tangan yang sudah dipublikasi. Kemudian sistem akan mengambil data berdasarkan *input* tersebut. Lalu sistem melakukan *hash* pada data yang diambil sehingga didapatkan *message digest.* Sedangkan *input digital signature* akan dibuka dengan menggunakan kunci publik sehingga akan didapat *message digest* juga. Sehinggah apabila *message digest*-nya tidak sama, maka sistem akan mengembalikan nilai *error*. Apabila *message digest*-nya sama maka data dan *digital signature* tersebut valid lalu sistem akan mengembalikan beberapa nilai yang terkait dengan data tersebut seperti pemilik tanda tangan, tanda tangan, perihal, dan waktu publikasi.

4. Riwayat Tanda Tangan



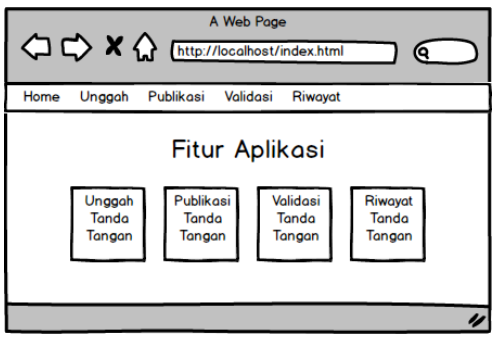
Gambar 7. *Flowchart* Riwayat Tanda Tangan

Berdasarkan *flowchart* pada gambar di atas, proses dimulai dengan *smart contract* menerima *input* berupa akun Metamask pengguna. Kemudian *smart contract* akan mengambil data berdasarkan *input* tersebut berupa riwayat publikasi pengguna yang nantinya akan ditampilkan.

D.3. Rancangan Interface

Berikut merupakan rancangan *interface* dari aplikasi *e-sign* berbasis *blockchain* Ethereum.

1. Home



Gambar 8. Halaman *Home*

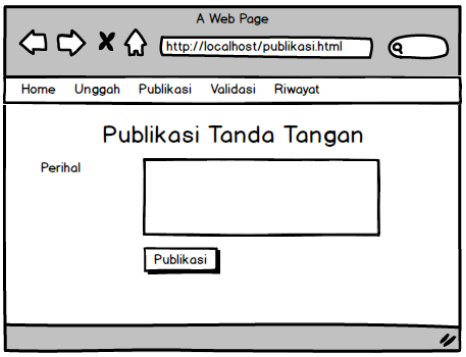
Pada halaman *Home*, pengguna dapat melihat fitur-fitur yang disediakan oleh sistem yaitu publikasi, validasi, dan riwayat tanda tangan.

2. Unggah

Gambar 9. Halaman Unggah

Pada halaman Unggah, pengguna dapat mengunggah tanda tangan yang ingin dijadikan tanda tangan elektronik atau *e-sign*.

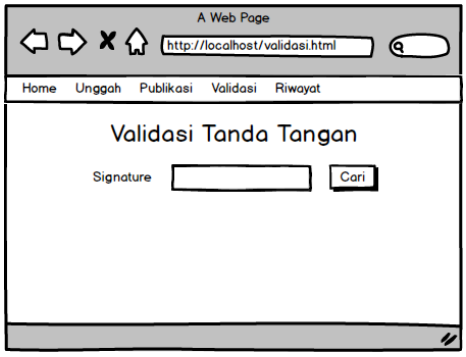
3. Publikasi



Gambar 10. Halaman Publikasi

Pada halaman publikasi tanda tangan, pengguna dapat mempublikasi tanda tangan ke dalam jaringan *blockchain* dengan hanya mengisi *form* perihal.

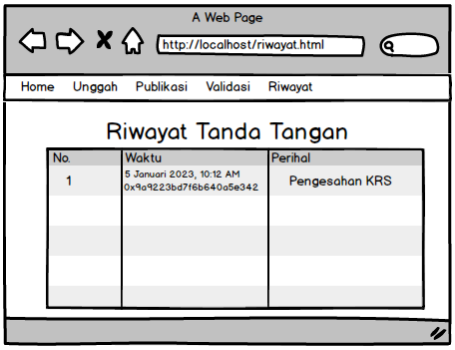
4. Validasi



Gambar 11. Halaman Validasi

Pada halaman validasi tanda tangan, pengguna dapat melakukan validasi terhadap suatu tanda tangan dengan cara mengisi *form* tersebut dengan *signature* dari tanda tangan yang telah di publikasi.

5. Riwayat



Gambar 12. Halaman Riwayat

Pada halaman riwayat tanda tangan, pengguna dapat melihat riwayat dari publikasi tanda tangan. Informasi riwayat berupa waktu publikasi, *signature*, dan perihal tanda tangan.

D.4. Implementasi

Dilakukan pembuatan aplikasi website *e-sign* berbasis *blockchain* Ethereum dengan menggunakan *smart contract* sebagai aturan dalam melakukan transaksi, Ganache sebagai *blockchain* lokal sekaligus penyimpanan terdesentralisasi, dan web3.js yang berfungsi sebagai penghubung antara *smart contract* dengan *front end* aplikasi.

D.5. Pengujian

Pengujian pada sistem dilakukan dalam empat tahap, yaitu *unit testing smart contract*, *blackbox testing*, mengakses sistem dengan komputer lain pada jaringan lokal dan analisis berdasarkan aspek dasar blockchain yaitu *immutable* atau tidak dapat diubah, transparan yaitu seluruh *node* dapat melihat semua transaksi yang ada pada *block*, dan terdesentralisasi yaitu setiap *node* menyimpan data yang sama.

E. Dokumentasi Laporan

Setelah melakukan pengujian sistem dan berjalan dengan semestinya, maka selanjutnya akan didokumentasikan dan diambil kesimpulan berdasarkan dokumentasi tersebut. Dari kesimpulan yang didapatkan akan digunakan sebagai acuan untuk pengembangan sistem selanjutnya.

IV. HASIL DAN PEMBAHASAN

Pada bagian hasil dan pembahasan ini akan dibahas hasil dari penelitian yang telah dirancang sebelumnya yang meliputi implementasi dan pengujian.

A. Implementasi

Setelah tahap perancangan selesai dibuat, tahap selanjutnya adalah tahap implementasi dari sistem yang akan dibangun. Tahap implementasi ini akan dibagi menjadi dua proses yaitu impelemntasi *smart contarct* dan implementasi *interface.*

A.1. Implementasi Smart Contract

Pada tahap implementasi *smart contract*, hal pertama yang dilakukan adalah membuat fungsi-fungsi utama yang akan digunakan pada setiap fitur. Adapun fungsi-fungsi tersebut sebagai berikut:

1. Fungsi *setTtd()* dan *getTtd()*

   Fungsi *setTtd()* merupakan fungsi yang digunakan untuk menyimpan berkas tanda tangan pengguna ke dalam jaringan *blockchain*. Parameter yang dibutuhkan pada fungsi ini yaitu *cid* dari berkas tanda tangan yang telah diunggah pada Pinata IPFS dan *address* akun Metamask pengguna. Sedangkan *getTtd()* berfungsi untuk mengambil data tanda tangan pengguna. Parameter yang diperlukan pada fungsi ini yaitu *address* dari akun Metamask pengguna.
2. Fungsi *setDataTtd()* dan *getDataTtd()*

   Fungsi *setDataTtd()* merupakan fungsi yang digunakan untuk menyimpan sebuah tanda tangan elektronik beserta data yang berkaitan dengan tanda tangan tersebut seperti pemilik tanda tangan, perihal, *timestamp*, dan *digital signature*. Parameter yang diperlukan untuk fungsi ini yaitu *digital signature*, perihal, *cid* berkas tanda tangan, *timestamp*, *address* akun Metamask pengguna, dan *cid* tanda tangan yang telah digabungkan dengan kode QR. Sedangkan *getDataTtd()* berfungsi untuk mengambil data-data yang berkaitan dengan tanda tangan tersebut.
3. Fungsi *verify()*

   Fugnsi *verify()* merupakan sebuah fungsi yang digunakan untuk memvalidasi sebuah tanda tangan elektronik. Parameter yang diperlukan pada fungsi ini yaitu *address* akun Metamask pengguna, perihal, *cid* berkas tanda tangan, *timestamp*, dan *signature*. Nilai kembalian dari fungsi ini yaitu *true* atau *false*. Jika nilai kembaliannya *true*, maka tanda tangan tersebut valid, jika *false* maka tanda tangan tersebut tidak valid.
4. Fungsi *setRiwayat()* dan *getRiwayat()*

   Fungsi *setRiwayat()* merupakan sebuah fungsi yang digunakan untuk menyimpan riwayat dari sebuah tanda tangan yang telah dipublikasi. Parameter yang diperlukan pada fungsi ini yaitu *address* akun Metamask pengguna, *timestamp*, *signature*, dan perihal. Sedangkan *getRiwayat()* berfungsi untuk mengambil data-data riwayat dari seorang pengguna.

A.2. Implementasi Interface

Interface merupakan antarmuka yang akan berinteraksi secara langsung dengan pengguna sistem. Dalam implementasi *interface*, telah dikembangkan berdasarkan perancangan yang telah dibuat sebelumnya. Berikut merupakan implementasi dari perancangan sebelumnya:

1. Halaman *Home*

Gambar 13. Implementasi Halaman *Home*

Gambar 13 merupakan implementasi dari halaman *home*, pada halaman ini pengguna dapat melihat fitur-fitur yang disediakan oleh sistem yaitu publikasi, validasi, dan riwayat tanda tangan.

2. Halaman Unggah

Gambar 14. Implementasi Halaman Unggah

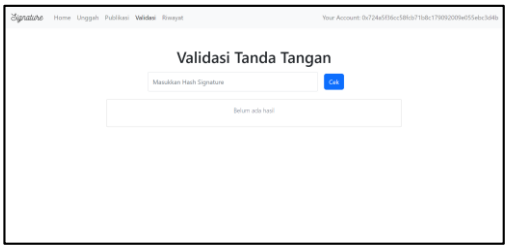
Gambar 14 merupakan implementasi dari halaman unggah, pada halaman ini pengguna dapat mengunggah berkas tanda tangan yang ingin dijadikan tanda tangan elektronik atau *e-sign*. Berkas tanda tangan yang diunggah nantinya akan disimpan ke dalam jaringan *blockchain*.

3. Halaman Publikasi

Gambar 15. Implementasi Halaman Publikasi

Gambar 15 merupakan implementasi dari halaman publikasi, pada halaman ini pengguna dapat mempublikasi tanda tangan ke dalam jaringan *blockchain*. Sebelum menggunakan fitur ini, pengguna diharuskan mengisi *form* perihal dan telah mengunggah tandan tangan pada fitur sebelumnya yaitu unggah tanda tangan.

4. Halaman Validasi



Gambar 16. Implementasi Halaman Validasi

Gambar 16 merupakan implementasi dari halaman validasi, pada halaman ini pengguna dapat melakukan validasi terhadap suatu tanda tangan dengan cara mengisi *form* tersebut dengan *signature* dari tanda tangan yang telah di publikasi.

5. Halaman Riwayat

Gambar 17. Implementasi Halaman Riwayat

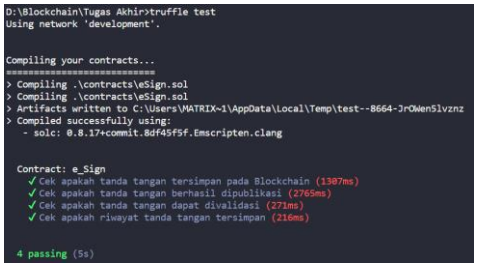
Gambar 4.11 merupakan implementasi dari halaman riwayat, pada halaman ini pengguna dapat melihat riwayat dari publikasi tanda tangan. Informasi riwayat berupa waktu publikasi, *signature*, dan perihal tanda tangan.

B. Pengujian

Pengujian sistem bertujuan untuk menguji coba sistem yang telah selesai dibuat pada tahap sebelumnya yaitu implementasi. Pengujian pada sistem dilakukan dalam empat tahap, adapun keempat tahapan tersebut sebagai berikut:

B.1. Pengujian Unit Testing Smart Contract

Pengujian ini bertujuan untuk memastikan bahwa setiap unit fungsi yang ada pada *smart contract* dapat berjalan dengan baik. Pengujian dilakukan pada *smart contract* yang belum di *deploy* ke jaringan *blockchain*. Unit fungsi yang diuji adalah fungsi-fungsi yang telah dijelaskan pada subbab sebelumnya yaitu impelementasi *smart contract* yang merupakan fungsi-fungsi utama dalam sistem yang dibangun. Adapun hasil dari pengujian terdapat pada gambar 4.12 berikut:



Gambar 18. Hasil Unit Testing Smart Contract

Pada Gambar 18, terdapat 4 poin pengujian yang mewakili setiap fitur yang dibangun pada sistem yaitu unggah, publikasi, validasi, dan riwayat tanda tangan. Berdasarkan Gambar 18, keempat poin tersebut sudah lulus dari pengujian *unit smart contract* yang ditandakan oleh kata "4 passing". Sehingga dapat disimpulkan bahwa seluruh unit fungsi telah berjalan dengan baik.

B.2. Blackbox Testing

Pengujian ini bertujuan untuk menguji fungsionalitas dari sistem yang telah dibuat tanpa mengetahui detail mengenai implementasi, struktur kode, dan jalur internal. Pengujian ini dapat mengukur kemampuan sistem dalam memenuhi kebutuhan pengguna serta mengetahui kesalahan-kesalahan pada sistem berdasarkan *input* dan *output* yang dihasilkan. Berikut hasil dari pengujian *blackbox testing*:

TABEL I. HASIL PENGUJIAN UNGGAH TANDA TANGAN

Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian
Mengunggah tanda tangan	Sistem memberikan pesan bahwa tanda tangan berhasil diunggah.	Sesuai

TABEL II. HASIL PENGUJIAN PUBLIKASI TANDA TANGAN

Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian
Mengisi *form* publikasi tanda tangan dengan lengkap	Sistem membeirkan pesan bahwa tanda tangan telah terpublikasi ke jaringan *blockchain* dan memberikan tanda tangan beserta kode QR dan *signature*	Sesuai
Tidak mengisi *form* publikasi tanda tangan	Sistem memberikan pesan bahwa *form* harus diisi	Sesuai
Melakukan publikasi sebelum mengunggah tanda tangan	Sistem memberikan pesan bahwa pengguna harus mengunggah tanda tangan terlebih dahulu	Sesuai

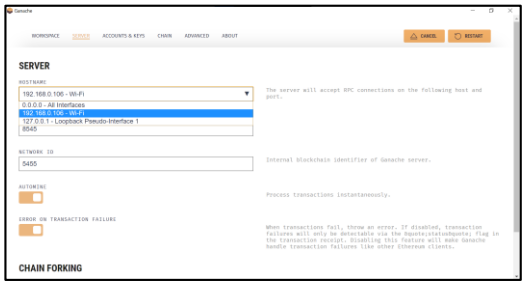
TABEL III. HASIL PENGUJIAN VALIDASI TANDA TANGAN

Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian
Mengisi kolom *search* dengan *signature* yang valid	Sistem menampilkan pemilik tanda tangan, tanda tangan, perihal, dan waktu publikasi.	Sesuai
Mengisi kolom *search* dengan *signature* yang tidak valid	Sistem memberikan pesan bahwa *signature* tidak valid.	Sesuai

B.3. Mengakses Sistem pada Jaringan Lokal

Pada tahap ini, sistem akan diuji dengan cara diakses oleh komputer lain pada jaringan lokal. Untuk dapat diakses oleh komputer lain, ada beberapa pengaturan yang dilakukan pada komputer *server* dan komputer *client*. Berikut pengaturan yang dilakukan pada kedua komputer tersebut:

1. Komputer *Server*



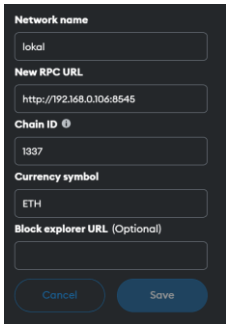
Gambar 19. Pengaturan *Hostnae* pada *Server* Ganache

Langkah pertama yang dilakukan yaitu mengubah "HOSTNAME" *server* pada aplikasi ganache dengan menggunakan "192.168.0.106 – WIFI" agar ganache dapat diakses oleh komputer lain dengan mengakses URL tersebut.

Gambar 20. Pengaturan *Host* pada Truffle Config

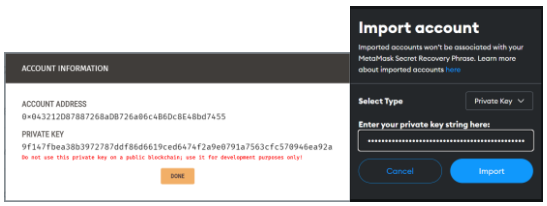
Langkah kedua yaitu mengubah host pada *truffle-config.js* menjadi "192.168.0.106" berdasarkan URL yang ada pada ganache.

2. Komputer *Client*



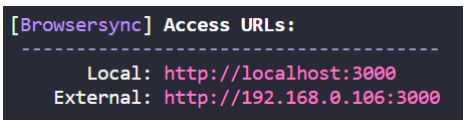
Gambar 21. Pengaturan *Network* pada Metamask

Langkah pertama yang dilakukan pada komputer client yaitu menambah *network* pada metamask dengan komposisi seperti pada Gambar 4.15. Penambahan *network* ini berfungsi agar metamask pada komputer *client* dapat mengakses ganache yang ada pada komputer *server*.

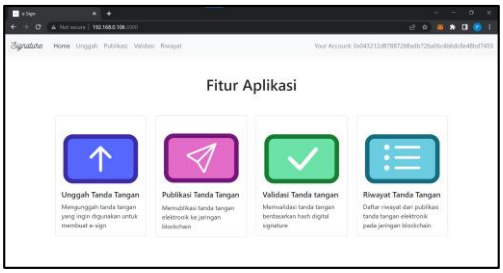


Gambar 22. *Import* Akun Metamask

Langkah kedua yaitu melakukan *import* akun pada metamask komputer *client* berdasarkan akun yang ada pada aplikasi ganache komputer *server* agar komputer *client* dapat memiliki saldo untuk melakukan transaksi pada sistem.



Gambar 23. URL Sistem dari Komputer *Server*



Gambar 24. Pengaksesan Sistem oleh Komputer *Client*

Ketika *server* telah dijalankan pada komputer *server*, maka komputer *client* dapat mengakses sistem dengan mengunjungi URL yang diberikan oleh komputer *server*. Sehingga dapat disimpulkan bahwa sistem yang dibangun dapat diakses oleh komputer lain yang berada pada jaringan yang sama.

B.4. Berdasarkan Aspek Dasar Blockchain

Pengujian ini merupakan analisis yang didasarkan pada sifat dasar *blockchain* yaitu *immutable* atau tidak dapat diubah, transparan yaitu seluruh *node* dapat melihat semua transaksi yang ada pada *block*, dan terdesentralisasi yaitu setiap *node* menyimpan data yang sama. Berikut hasil analisis berdasarkan aspek dasar *blockchain* tersebut:

1. *Immutable*

   Immutable merupakan sebuah sifat *blockchain* yang berarti tidak dapat diubah. Setelah suatu blok ditambahkan ke dalam rangkaian *blockchain*, tidak mungkin untuk mengubah atau menghapus blok tersebut. Hal ini disebabkan karena blok tersebut terhubung secara berkelanjutan dengan blok

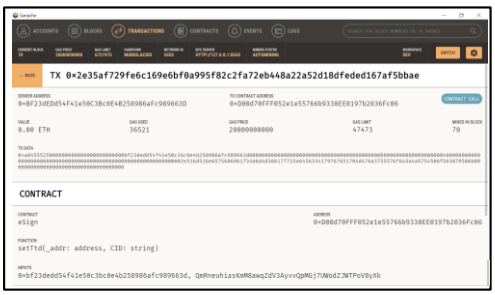
sebelumnya dan blok sesudahnya. Apapun perubahan kecil yang dilakukan pada blok tersebut, baik itu dalam bentuk modifikasi atau penghapusan, akan menyebabkan perubahan pada semua blok yang ada dalam rangkaian. Tindakan tersebut dianggap tidak mungkin dapat dilakukan [20].

Gambar 25. Sifat *Immutable*

Gambar 25 menunjukkan beberapa detail komponen transaksi dalam *blockchain* yaitu *transactionHash, transactionIndex, blockHash,* dan *blockNumber*. Selain itu pada setiap transaksinya, *blockchain* Ethereum selalu melakukan *hash* dan *digital signature*. Sehingga terdapat tujuh lapisan keamanan dalam suatu blok, yang memerlukan waktu dan sumber daya yang besar untuk membuka dan mengubah data di dalamnya. Keamanan pada tujuh tingkat lapisan tersebut dapat diandalkan berkat sifat *immutability* dari *blockchain* [21]. Jika dibandingkan dengan tanda tangan elektronik biasa, tanda tangan elektronik yang menggunakan teknologi *blockchain* terlihat lebih aman dikarenakan menggunakan mekanisme enkripsi, sehingga keamanan data pada jaringan *blockchain* dapat terjamin. Sedangkan tanda tangan elektronik biasa tidak memiliki sistem keamanan.

2. Transparan

Setiap transaksi yang terjadi dapat diketahui oleh semua orang yang berpartisipasi dalam sistem, dan semua peserta memiliki akses untuk mengetahui siapa yang melakukan tindakan apa pada saat yang bersamaan. Sehingga setiap orang bisa mengetahui dan memverifikasinya demi memberikan akunt-abibilitas dan transparansi kepada pihak-pihak yang melakukan transaksi.

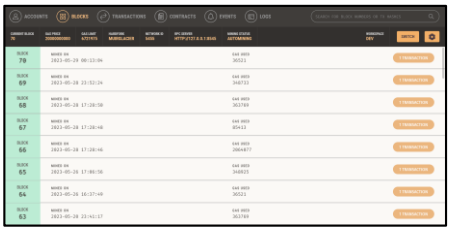


Gambar 26. Sifat Transparan

Berdasarkan Gambar 26 menunjukkan bahwa setiap transaksi yang telah dilakukan oleh seseorang, semua orang dapat melihat ke dalam isi dari sebuah blok transaksi. Sehingga dapat menimbulkan rasa saling percaya dan hal tersebut dapat membuktikan bahwa *blockchain* bersifat transparan. Jika dibandingkan dengan tanda tangan elektronik biasa, tanda tangan elektronik yang menggunakan teknologi *blockchain* dapat divalidasi dikarenakan oleh sifat transparansi dari *blockchain.* Sedangkan tanda tangan elektronik biasa tidak dapat divalidasi.

3. Terdesentralisasi

Teknologi *blockchain* mengambil prinsip dasar dari sistem terdesentralisasi, di mana setiap informasi yang disimpan dalam jaringan *blockchain* akan didistribusikan ke semua partisipan jaringan. Dalam hal keamanan, tidak ada satu entitas pun yang memiliki kendali mutlak atas data atau transaksi yang terjadi di jaringan *blockchain*. Setiap *node* atau *validator* dalam jaringan memiliki salinan data yang identik. Dengan demikian, teknologi *blockchain* memungkinkan terjadinya transaksi yang aman dan efisien tanpa harus bergantung pada suatu organisasi atau entitas berpengaruh yang mengatur aliran transaksi data [22].



Gambar 27. Sifat Terdesentralisasi

Pada sifat terdesentralisasi, seluruh *node* memiliki salinan yang sama berupa blok-blok seperti pada Gambar 4.15. Sehingga seluruh *node* dapat mengakses seluruh data yang tersimpan pada jaringan *blockchain.* Dari sisi penyimpanan, jika dibandingkan dengan tanda tangan elektronik biasa, penyimpanan tanda tangan elektronik yang menggunakan teknologi *blockchain* bersifat desentralisasi. Sehingga apabila terjadi kerusakan data pada suatu *node,* maka tidak akan mengubah atau mempengaruhi transaksi data pengguna lain. Sedangkan pada penyimpanan yang bersifat sentral, apabila seseorang memiliki akses ke *database* tersebut maka kerusakan seluruh data dapat terjadi.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, didapat beberapa kesimpulan sebagai berikut:

1. Pengimplementasian tanda tangan elektronik pada jaringan privat *blockchain* dapat dilakukan dengan menggunakan jaringan *blockchain* Ethereum lokal yaitu Ganache sebagai pengganti *database*.
2. Berdasarkan pengujian yang telah dilakukan, setiap unit fungsi pada *smart contract* dapat berjalan dengan baik dan secara fungsionalitas seluruh fitur pada sistem berfungsi sesuai dengan yang direncanakan.

3. Sistem yang telah dibuat dapat diakses oleh komputer lain yang berada pada jaringan yang sama dengan komputer *server*. Sistem dapat diakses dengan cara mengatur URL yang ada pada Ganache dan *host* pada *truffle config* di komputer *server*. Sedangkan pada komputer *client*, hanya perlu melakukan *add network* dan *import* akun pada Metamask. Sistem kemudian dapat diakses dengan mengunjungi URL yang diberikan oleh komputer server ketika server telah dijalankan.

B. Saran

Adapun saran yang dapat diberikan agar sistem yang telah dibangun menjadi lebih baik lagi adalah diharapkan untuk menggunakan jaringan *blockchain* publik sehingga dapat diketahui bagaimana performa dari *website* pada jaringan publik.

DAFTAR PUSTAKA

[1] A. N. El Izzah and W. Sugandha, "Penggunaan Tanda Tangan Elektronik Dalam Penyelenggaraan E-Government Guna Mewujudkan Pelayanan Publik Yang Efisien," *J. Law, Soc. Islam. Civiliz.*, vol. 9, no. 1, p. 1, 2021, doi: 10.20961/jolsic.v9i1.52836.

[2] Privy, "Tanda Tangan Elektronik VS Tanda Tangan Digital," Privy, 2021. https://blog.privy.id/digital-signature-adalah/

[3] W. Swastika, H. W. Santo, and O. H. Kelana, "Rancang Bangun Website Akademik Dengan Penyimpanan Sertifikat Digital Menggunakan Teknologi Blockchain," J. Teknol. Inf. dan Ilmu Komput., vol. 9, no. 1, pp. 33–40, 2022, doi: 10.25126/jtiik.202293645.

[4] S. Khan, A. Arshad, G. Mushtaq, A. Khalique, and T. Husein, "Implementation of Decentralized Blockchain E-voting," EAI Endorsed Trans. Smart Cities, vol. 4, no. 10, p. 164859, 2020, doi: 10.4108/eai.13-7-2018.164859.

[5] M. Zakie, "Implementasi Teknologi Blockchain Menggunakan Smart Contract Pada E-Voting," Universitas Islam Negeri Suska Riau, 2021.

[6] H. Vranken, "Sustainability of bitcoin and blockchains," Curr. Opin. Environ. Sustain., vol. 28, pp. 1–9, 2017, doi: 10.1016/j.cosust.2017.04.011.

[7] T. Harlian, Y. Purwanto, and M. F. Ruriawan, "Implementasi Blockchain Untuk Pendataan Dokumen Digital," eProceedings Eng., vol. 9, no. 3, pp. 1076–1079, 2022.

[8] Rifa Hanifatunnisa and Muhammad Ismail, "Desain dan Implementasi Sistem Pencatatan Pemungutan Suara dengan Teknologi Blockchain pada Jaringan Peer-to-Peer," J. Nas. Tek. Elektro dan Teknol. Inf., vol. 9, no. 4, pp. 354–364, 2020, doi: 10.22146/jnteti.v9i4.648.

[9] S. Chen, J. Zhang, R. Shi, J. Yan, and Q. Ke, "A comparative testing on performance of blockchain and relational database: Foundation for applying smart technology into current business systems," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10921 LNCS, no. September, pp. 21–34, 2018, doi: 10.1007/978-3-319-91125-0_2.

[10] R. M. Awangga, R. M. Komaran, and M. J. S. Wicaksana, Menerapkan Roblox di Metaverse, 1st ed. Bandung: Penerbit Buku Pedia, 2022.

[11] M. Khoirul Umam, "Perdagangan Ethereum Di Indodax Exchange Dalam Persfektif Syariah," ISTITHMAR J. Pengemb. Ekon. Islam, vol. 3, no. 2, pp. 169–192, 2020, doi: 10.30762/itr.v3i2.2050.

[12] Sugiyatno and P. D. Atika, "Digital Signature Dengan Algoritma Sha-1 Dan Rsa Sebagai Autentikasi," J. Cendikia, vol. 16, no. 2, pp. 74–83, 2018.

[13] M. M. Amin, "Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks," Pseudocode, vol. 3, no. 2, pp. 129–136, 2017, doi: 10.33369/pseudocode.3.2.129-136.

[14] K. Aryasa and Y. T. Paulus, "Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java," Creat. Inf. Technol. J., vol. 1, no. 1, p. 57, 2015, doi: 10.24076/citec.2013v1i1.10.

[15] S. Adilah, R. Rumani M, M. W. Paryasto, P. S1, and S. Komputer, "Implementasi Kriptosystem Menggunakan Metode Algoritma Ecc Dengan Fungsi Hash Sha-256 Pada Sistem Ticketing Online," eProceedings Eng., vol. 4, no. 3, pp. 4138–4146, 2017, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/5471

[16] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 1, no. 9, pp. 803–812, 2017, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/247

[17] E. Cahyo Prabowo and I. Afrianto, "Penerapan Digital Signature Dan Kriptografi PadaOtentikasi Sertifikat Tanah Digital - Teknik Informatika Universitas Komputer Indonesia," J. Ilm. Komput. dan Inform., vol. 6, no. 2, 2017.

[18] R. Gupta, B. Jha, A. K. Shukla, A. Raj, and S. Sultana, "Secure and Decentralized Smart Elections," vol. 22, no. 4, pp. 52–57, 2020, doi: 10.9790/0661-2204015257.

[19] "Apa Itu IPFS Dan Kegunaannya Di NFT Project - Diginews.id," April 1, 2022. https://diginews.id/apa-itu-ipfs-dan-kegunaannya-di-nft-project/

[20] T. P. Utomo, "Implementasi Teknologi Blockchain Di Perpustakaan: Peluang, Tantangan Dan Hambatan," Bul. Perpust., vol. 4, no. 2, pp. 173–200, 2022.

[21] D. Pratama, Yulfitno Wingga, Kurniadi, "IMPLEMENTASI BLOCKCHAIN DALAM APLIKASI PEMILU," vol. 02, no. 02, 2021.

[22] F. A. Saputra and U. K. Indonesia, "Teknologi Blockchain Dalam Menjaga Keamanan Data," no. April, 2023.