

**IMPLEMENTASI TEKNOLOGI BLOCKCHAIN PADA WEBSITE E-VOTE
MENGUNAKAN BAHASA PEMROGRAMAN PHYTON**

***BLOCKCHAIN TECHNOLOGY IMPLEMENTATION ON E-VOTE WEBSITE
USING PYTHON PROGRAMMING LANGUAGE***

Muhammad Reza Pahlevi¹, Siti Madinah Ladjamuddin²

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Informasi
Institut Sains dan Teknologi Nasional
Jl. Moh. Kahfi II, Bhumi Srengseng Indah, Jagakarsa, Jakarta Selatan 12640
Telp. (021) 7874647, Fax. (021) 7866955

¹muhammadreza1261@gmail.com, ²citymadinah07@istn.ac.id

ABSTRAKSI

Pemilihan umum adalah hal yang sering menjadi polemik ketika hasil dari *voting* tersebut terasa janggal. Banyak kasus dimana kecurangan dalam sebuah pemilihan terjadi karena banyak kekurangan dalam sistem pemilihan yang diterapkan terutama dalam *electronic voting*. Dalam pelaksanaan pemilihan umum ini masih menggunakan metode konvensional sehingga tidak efektif, efisien, dan fleksibel serta harus menyediakan anggaran untuk pengadaan kertas dan juga tidak ramah lingkungan. Penerapan teknologi Blockchain dalam *website e-voting* merupakan salah satu pencegahan dalam kecurangan dan pembobolan sistem. Dengan menerapkannya akan menghasilkan data yang *real* dan tidak dimanipulasi. Dengan sistem *desentralized distributed database* yang diamankan dengan sistem enkripsi akan menjamin keamanan *database* dengan baik.

Kata Kunci : *Blockchain, electronic voting, voter, python, website*

ABSTRACTION

General elections are things that often become polemic when the results of the voting seem odd. Many cases where fraud in an election occurs due to many deficiencies in the electoral system that is applied, especially in electronic voting. In the implementation of this general election still using conventional methods so that it is not effective, efficient, and flexible and must provide a budget for paper procurement and also not environmentally friendly. The application of Blockchain technology on e-voting websites is one of the prevention of fraud and system breaches. By implementing it will produce real data and not manipulated. With a decentralized distributed database system that is secured with an encryption system, it will ensure the security of the database properly.

Keywords : *Blockchain, electronic voting, voter, python, website*

1. PENDAHULUAN

Electronic voting (e-Voting) pertama kali diperkenalkan oleh David Shaumm pada awal 1980. Sistem yang digunakan adalah dengan *cryptography-key* yang membantu para *voter* untuk tetap tidak terdeteksi. Estonia adalah negara yang pertama kali menggunakan *electronic voting* hanya menggunakan internet dan kartu tanda penduduk elektronik (e-KTP). Negara selanjutnya yang mengimplementasikan *electronic voting* adalah Norwegia. Sistem yang dibuat mirip seperti yang dimiliki Estonia, tetapi terpaksa tidak diteruskan karena banyak pihak yang takut akan keamanan dari sistem itu. Washington D. C. juga mengembangkan *electronic voting* pada tahun 2010. Tetapi banyak sekali masalah keamanan pada saat melakukan pengujian pada sistemnya. Sehingga proyek tersebut tidak pernah diimplementasikan.

Blockchain merupakan teknologi dasar dari sebuah desain arsitektur *cryptocurrency Bitcoin*

yang diciptakan oleh Satoshi Nakamoto pada tahun 2008. Ini merupakan bentuk dari *distributed database* yang mana berisi dari transaksi- transaksi yang disimpan dalam sebuah *block* data. Dengan menggunakan teknologi penyimpanan dan keamanan data seperti ini akan membuat sistem pemilihan umum secara *online* atau *e-voting* akan terjaga keamanan dan transparansi data yang ada pada sistem tersebut.

Dengan acuan terkait dari jurnal IMPLEMENTASI BLOCKCHAIN: STUDI KASUS E-VOTING Satria Damai Kurnia Hu, Henry Novianus Palit, Andreas Handojo Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra dan Jurnal APLIKASI VOTING ONLINE DENGAN MENGGUNAKAN TEKNOLOGI BLOCKCHAIN Ahmad Fajar Prasetyo, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Oleh karena itu, sistem dan aplikasi yang akan diciptakan ini akan menjawab beberapa

kebutuhan pemungutan suara dengan implementasi dari teknologi *blockchain*. Sistem tersebut akan menjawab *transparency*, yaitu data yang disimpan bersifat terbuka untuk publik, sehingga meningkatkan keadilan dan kebenaran. *Anonymity*, yaitu hanya *voter* itu sendiri yang tahu informasi mengenai *vote* dan semua *ballot* yang terkumpul tidak ada hubungannya dengan *voter*. *Dependability*, yaitu setiap *vote* akan dihitung dan tidak dapat diganti, digandakan ataupun dihapus, serta mengeluarkan hasil yang dapat dipercaya. *Eligibility*, yaitu hanya user yang terverifikasi dan memiliki hak suara yang dapat membuat *ballot* dan melakukan *vote*. *Verifiability*, yaitu sistem bersifat terbuka untuk dapat diperiksa kebenarannya dari prosedur sistem hingga hasil yang dikeluarkan. Sehingga dengan begitu fungsi dari aplikasi e-vote ini sebagai media penyelenggara pemilihan umum berbasis website akan dapat tercapai.

Berdasarkan dari uraian latar belakang di atas, dirumuskan masalah yang akan dibahas dalam penelitian, yaitu sebagai berikut :

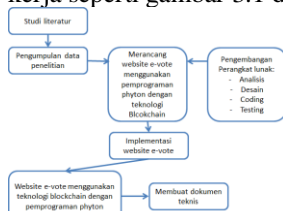
1. Bagaimana menerapkan sistem blockchain pada website e-vote
2. Bagaimana membangun website dengan bahasa pemrograman python dan dengan framework *django* dan *pycryptodome*.

Agar penelitian lebih terarah, maka diberi batasan masalah sebagai berikut: Aplikasi yang dibangun berbasis website dengan menggunakan bahasa pemrograman *Python* dan database *MySQL*; Database disimpan menggunakan sistem *blockchain*; Metode enkripsi database menggunakan *SHA3-256*; Validasi dilakukan oleh pihak admin atau pengawas; Data voter yang akan dimasukan adalah data yang terverifikasi pihak lain; Sistem tidak memeriksa dan memproses data diluar aplikasi.

2. METODOLOGI PENELITIAN

Desain Penelitian

Desain penelitian akan digambarkan pada kerangka kerja seperti gambar 3.1 dibawah ini :



Gambar 3.1 Kerangka Kerja Penelitian

Pada gambar 3.1 diatas dijelaskan bagaimana perancangan aplikasi dimulai, yaitu :

1. Studi literatur, pada tahap ini penulis mengumpulkan data dari berbagai macam sumber untuk menjadi refrensi penulisan dan contoh aplikasi yang akan dibuat, kebanyakan sumber berasal dari jurnal dan internet.

2. Pengumpulan data penelitian, pada tahap ini penulis mulai mengumpulkan data yang sudah dicari pada berbagai sumber, dan merangkumnya untuk melanjutkan pada tahap rancangan dan penulisan.

3. Merancang aplikasi, pada tahap ini penulis mulai merancang aplikasi, pada awal mendesain pada mockup, coding hingga hasil jadi, tahap ini didukung oleh metode waterfall.

4. Pengembangan perangkat lunak, pada tahap merancang pengembangan perangkat lunak menggunakan metode waterfall yang akan dijelaskan pada metode penelitian sub bab 2.

5. Implementasi, pada tahap ini penulis telah mengimplementasikan website ini dengan cara menggunakannya dan mencoba melihat kekurangan pada aplikasi sebelum pada tahap hasil akhir.

6. Pada tahap ini, website yang penulis rancang sudah jadi dan siap digunakan, aplikasi juga terus dikembangkan selaras dengan penggunaan aplikasi sesuai kebutuhan aplikasi, mulai dari fitur hingga tampilan.

7. Pada tahap dokumen teknis ini, penulis membuat laporan mengenai jalannya aplikasi seperti screenshot aplikasi dan pengujian blackbox.

Metode Penelitian

Adapun metode penelitian ini dapat digambarkan sesuai dengan yang telah dirancang pada gambar 3.1 yang terdiri atas :

Metode Pengumpulan Data

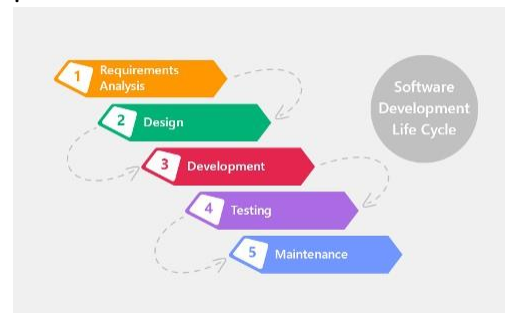
Metode pengumpulan data yang dipakai pada penelitian untuk menyusun aplikasi ini adalah metode studi literatur.

1. Metode literatur, mencari teori-teori pendukung mengenai aplikasi yang dibangun. Teori-teori tersebut dapat dicari melalui buku, jurnal ataupun dari internet.

2. Metode Pengembangan Perangkat Lunak

Model proses yang digunakan dalam pembangunan sistem ini adalah model

waterfall, yang melingkupi aktivitas sebagai berikut :



Gambar 3.2 Model *Waterfall*

Adapun langkah-langkahnya sebagai berikut:

1. Perencanaan (*planning*) Mengenai kebutuhan pengguna (*user's specification*), studi-studi

kelayakan (*feasibility study*) baik secara teknik maupun secara teknologi serta penjadwalan suatu proyek sistem informasi atau perangkat lunak. pada tahap ini pula, sesuai dengan kaskas (tool) yang penulis gunakan yaitu UML.

2. Analisis (*analysis*) Yaitu tahap dimana kita berusaha mengenai segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan *use case* diagram lebih lanjut, mengenai komponen-komponen sistem atau perangkat lunak, objek- objek, hubungan antar objek dan sebagainya.

3. Perancangan (*design*) Mencari solusi dari permasalahan yang didapat dari tahap analisis dan menetapkan form atau bentuk dari aplikasinya.

4. Implementasi
Mengimplementasikan perencanaan sistem ke situasi nyata yaitu dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (*pengkodean/coding*).

5. Pengujian (*testing*)
Menentukan apakah sistem atau perangkat lunak yang dibuat sudah sesuai dengan kebutuhan pengguna atau belum, jika belum, proses selanjutnya adalah bersifat iteratif, yaitu kembali ketahap-tahap sebelumnya. Dan tujuan dari pengujian itu sendiri adalah untuk menghilangkan atau meminimalisasi cacat program (*defect*) sehingga sistem yang dikembangkan benar-benar akan membantu para pengguna saat mereka melakukan aktivitas-aktivitasnya.

6. Pemeliharaan (*maintenance*)
Proses pemeliharaan sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil.

Alat dan Bahan Penelitian

Alat Penelitian

Pada penelitian ini digunakan alat penelitian berupa perangkat keras danperangkat lunak sebagai berikut :

- Komputer dengan spesifikasi :
- 1. Processor Intel core i3 , up to 2.53 Ghz
- 2. RAM 2gb up to 4gb
- 3. Harddisk 500Gb SATA
- 4. Monitor dengan resolusi 1366 x 768
- 5. Mouse keyboard

Perangkat Lunak :

- 1. Operating system : Windows 10 Pro 64-bit
- 2. Software pendukung : MySQL, Phyton 3.8, Django, PyCryptodome
- 3. Bahasa Pemrograman Phyton
- 4. *Software* utama : *pycharm*

Bahan Penelitian

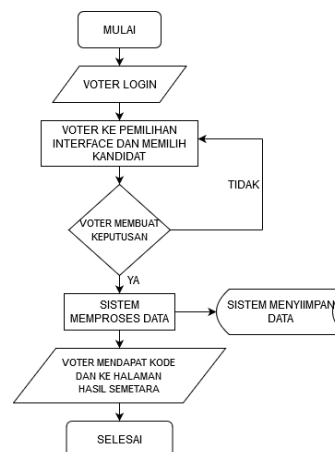
Bahan-bahan penelitian berupa data pemrograman berupa *website* dengan fitur keamanan *Blockchain* yang dimodifikasi agar dapat digunakan pada *website e-vote*. Data juga dapat

berupa modeling dari sistem *e-vote* yang sudah ada dan dikombinasikan. Data-data ini digunakan sebagai dasar kasus dalam pembuatan aplikasi ini.

Analisa dan Perancangan Aplikasi

Analisis dan Perancangan Sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian untuk mengidentifikasi dan mengevaluasi permasalahan sehingga dapat membangun system yang baik. Komponen-komponennya antara lain *flowchart*, *use case diagram* dan *diagram activity*.

Flowchart Diagram

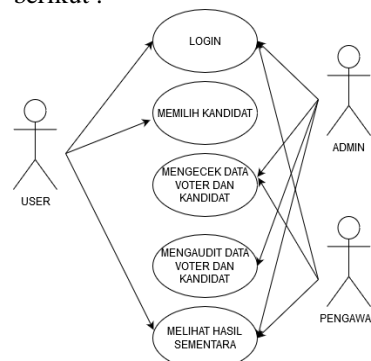


Gambar 3.3 Flowchart Diagram

Pada gambar 3.3 diatas, Proses dari sistem aplikasi yang akan berjalan nantinya yaitu dimulai dari login voter, lalu voter akan diarahkan ke menu *interface* pilihan kandidat, kemudian voter harus memilih salah satu dari kandidat, setelah itu sistem akan memproses data dari voter dan pilihan yang dipilih dan menyimpannya dengan sistem *Blockchain*, setelah itu voter akan mendapat kode transaksi dari sistem untuk validasi dan voter dapat melihat hasil sementara pada laman hasil sementara.

Use Case Diagram

Use case diagram pada aplikasi adalah sebagai berikut :

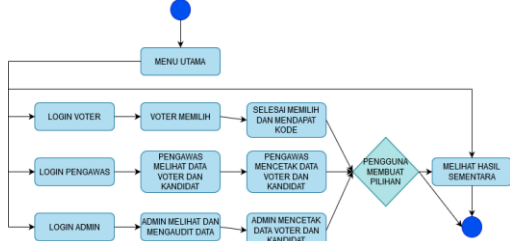


Gambar 3.4 Use Case Diagram

Pada gambar 3.4 diatas voter dapat melakukan login, memilih kandidat, dan juga melihat hasil sementara. Untuk pengawas dapat melakukan

login, mengecek dan verifikasi data dari voter dan juga melihat hasil sementara. Dan untuk admin dapat login, mengaudit data voter dan kandidat, lalu melihat hasil sementara.

Activity Diagram



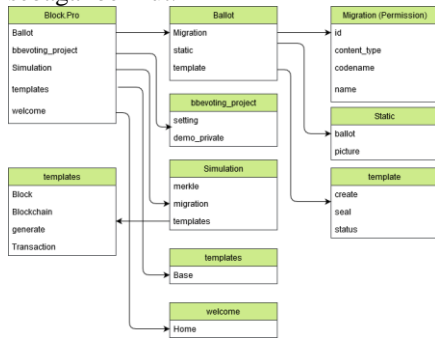
Gambar 3.5 Activity Diagram

Pada gambar 3.5 diatas, disajikan bagaimana aplikasi berjalan dari mulai menu utama, lalu pengguna dapat menggunakan fitur login sesuai apa status mereka, jika mereka voter maka mereka dapat memilih kandidat lalu mendapatkan kode, setelah itu voter bisa melihat hasil sementara dari voting.

Sedangkan jika status mereka admin, maka dapat melakukan login, mengaudit data voter dan kandidat, mencetaknya, dan dapat melihat hasil sementara. Berbeda dengan admin, pengawas hanya bisa melihat data voter dankadidat tanpa bisa merubah isinya, pengawas dapat mencetak data dan melihat hasil sementara juga.

Django Class Diagram

Class diagram dari aplikasi web e-vote ini adalah sebagai berikut:



Gambar 3.6 Class Diagram Django Project

Pada gambar 3.6 diatas disajikan bagaimana struktur dari project website e-vote ini dalam bentuk class diagram, project ini menggunakan framework django sebagai environment nya, dalam project ini terdapat struktur sebagai berikut:

1. Block.Pro adalah struktur utama dari website e-voting ini, isi dari project ini antara lain yaitu Ballot, bbevoting_project, simulation, templates, dan welcome.
2. Ballot disini adalah sebuah folder yang mengatur migration database, static yang berisi data gambar dari kandidat, dan picture, sebagai data gambar cadangan.
3. Migration (permission), adalah tempat dimana program untuk migration data voter yang berisi id,

codename, content_type, dan name.

4. Bbvoting_project, adalah tempat dimana settingan dari aplikasi django ini diatur seperti banyak data dan juga banyak block yang akan dibuat, dan berisi data private key yang akan dipakai sistem untuk enkripsi dan dekripsi data.

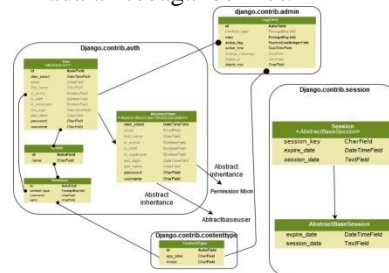
5. Simulation, adalah folder project dimana data program untuk merkle tree, migration dan templatnya dibuat disini, dan didalam template ada data website interface seperti halaman Block, Program Blockchain, Generating signature, dan halaman transaksi.

6. Templates, adalah suatu struktur halaman utama dari website yang didalamnya terdapat file berupa base untuk menu halaman utama.

7. Welcome, adalah page dari awal menu utama yang ditimpa dengan halaman templates, file ini adalah bagian dari halaman menu utama website itu sendiri.

Class Diagram SQLite

Class diagram dari database SQLite pada aplikasi ini adalah sebagai berikut:



Gambar 3.7 Class Diagram SQLite

Pada Gambar 3.7, disajikan gambar mengenai class diagram dari database sqlite yang digunakan dalam aplikasi website e-vote ini, isi dari class diagram sqlite ini adalah sebagai berikut:

1. Django.contrib.admin yaitu berisi database log.entry yang berisi data berupa id, content_type, user, action_flag, action_time, chane_message, object_id dan object_repr. Django contrib.admin ini berhubungan dengan django.contrib.auth dan django.contrib.contenttype.
2. Django.contrib.auth yaitu berisi database otentifikasi user dengan data id, date_joined, email, first_name, is_active, is_staff, is_superuser, last_login, last_name, password, dan username. Pada tahap ini, contribauth ini berhubungan dengan contenttype dan contribadmin.
3. Django.abstact.user berisi data user seperti pada contrib.auth,data yang dimasukan juga sama karena ini adalah hasil inheritance dari data sebelumnya.
4. Group, adalah database yang berisi otentifikasi dari grup data yang dimasukan dan diproses oleh sistem.
5. Permission, adalah database yang berisi data grup yang sedang diproses oleh sistem untuk melakukan permission data sebelum data diolah ke

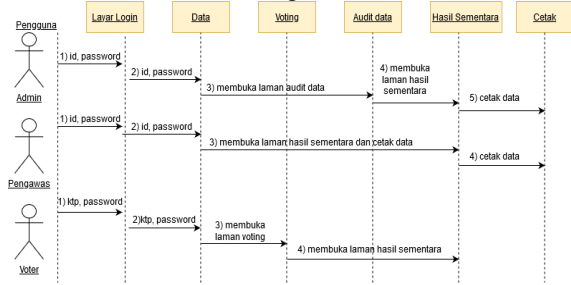
proses selanjutnya.

6. Django.contrib.contenttype, adalah data konten yang telah diverifikasi oleh sistem pada permission, dan data memiliki model dan label.

7. Django.contrib.session, adalah data dari session pengguna ketika melakukan login atau perpindahan halaman yang memerlukan status login atau diizinkan oleh sistem.

Sequence Diagram

Sequence diagram yang digunakan pada aplikasi website e-vote adalah sebagai berikut:



Gambar 3.8 Sequence Diagram

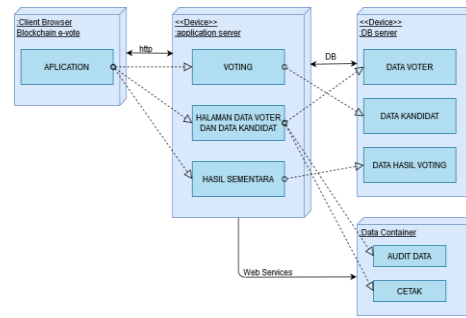
Gambar 3.8 diatas merupakan rancangan sequence diagram dari aplikasi e-vote tersebut, yang menunjukkan interaksi dari objek-objek dan memberi petunjuk komunikasi dari objek tersebut. Dalam interaksi yang terjadi pada aplikasi terjadi tiga level pengguna yaitu admin, pengawas, dan voter yang memiliki interaksi berbeda-beda pada objek. Interaksi yang terjadi yaitu sebagai berikut:

1. Admin, pada tahapan pengguna admin akan berinteraksi dengan layar login dengan tahapan data id dan password, lalu interaksi pada data yaitu data login untuk admin id dan password, lalu interaksi dengan audit data dengan membuka laman audit data setelah login, lalu interaksi selanjutnya dengan membuka hasil sementara dan terakhir interaksi dengan cetak data.

1. Pengawas, pada tahapan pengawas akan berinteraksi dengan layar login dengan tahapan data id dan password, lalu interaksi pada data yaitu data login untuk pengawas id dan password, lalu interaksi dengan hasil sementara dan terakhir interaksi dengan cetak data.

2. Voter, pada tahapan voter akan berinteraksi dengan layar login dengan tahapan ktp dan password, lalu interaksi dengan pada data dengan tahapan ktp dan password, lalu interaksi dengan voting dengan memilih kandidat, terakhir interaksi dengan halaman hasil voting sementara.

Deployment Diagram



Gambar 3.9 Deployment Diagram

Pada gambar 3.9 merupakan rancangan dari deployment diagram pada website e-vote. Pada dasarnya deployment diagram ini hanya untuk menggambarkan, menspesifikasikan, dan mendokumentasikan yang terjadi pada aplikasi web dan pada deviceny dan apa saja isinya. Penjelasan pada deployment diagram diatas adalah sebagai berikut:

1. Client Browser, pada blok ini aplikasi website e-vote berada pada client browser pengguna, aplikasi ini mempunyai relasi dengan device server melalui hosting http dan menghubungkan dengan laman voting, dengan laman data voter dan data kandidat, dan juga dengan laman hasil sementara.

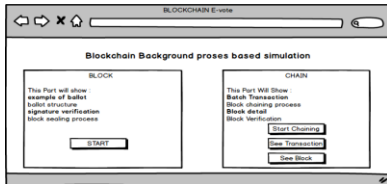
2. Device application server, pada blok ini berisi laman yang dapat dijalankan oleh server pada device, blok ini berisi proyek webitu sendiri, data berupa halaman dari website e-vote yaitu voting, halaman data voter dan kandidat, lalu halaman hasil sementara. Voting memiliki relasi dengan data kandidat, halaman data voter dan kandidat memiliki relasi dengan data voter, dan hasil sementara memiliki relasi dengan data hasil sementara yang ketiganya dihubungkan dengan koneksi database.

3. Device db server, adalah blok berisi database yang tersimpan pada device sehingga aplikasi memiliki akses terhadap database, data berisi data voter yang memiliki relasi dengan halaman voter dan kandidat, data kandidat yang memiliki relasi dengan laman voting, dan data hasil voting yang memiliki relasi dengan laman hasil sementara.

4. Data container, yaitu adalah fitur mengaudit data yang dimiliki oleh admin dan fitur cetak yang bisa digunakan oleh admin dan pengawas yang berada pada aplikasi dan terhubung pada web service.

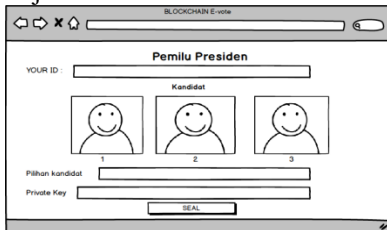
Perancangan Antar Muka Web Background Proses

Aplikasi web ini akan menampilkan semua proses pendataan dengan menggunakan metode blockchain dikarenakan aplikasi web utama tidak dapat melihat proses blockchain terjadi, maka aplikasi ini dibuat.



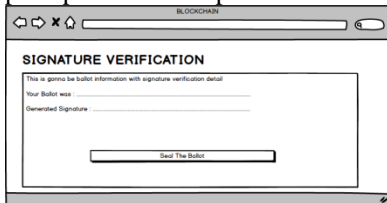
Gambar 3.23 Tampilan Menu Blockchain background proses

Pada gambar 3.23 diatas adalah tampilan utama dari web, disini pengguna dapat melihat bagaimana cara *block* dibuat, bagaimana proses *chaining* terjadi.



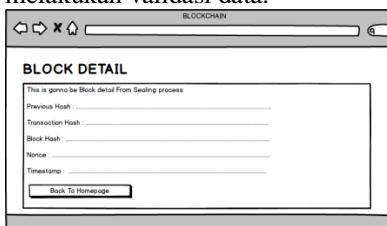
Gambar 3.24 Tampilan simulasi block dibuat

Pada gambar 3.24 diatas pengguna dapat melihat bagaimana cara web membuat *block* dari sebuah data yang dienkripsi dengan memasukan data yang telah dienkripsi tersebut, kandidat, dan *private key* pada pilihan lalu dapat submit datanya.



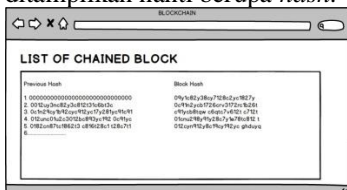
Gambar 3.25 Tampilan setelah data enkripsi dibuat

Gambar 3.25 menunjukkan bahwa data yang disimpan tadi sudah dienkripsi dan ada data tanda tangan yang merupakan kode yang diberikan pada *voter*, kode ini harus disimpan atau diingat untuk melakukan validasi data.



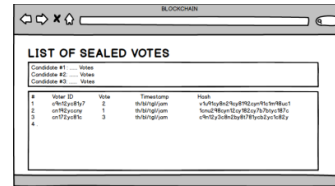
Gambar 3.26 Tampilan detail dari block yang dibuat

Gambar 3.26 diatas adalah hasil enkripsi dari proses membuat *block* yang berasal dari data *voter* yang sudah memilih kandidat, data yang akan ditampilkan nanti berupa *hash*.



Gambar 3.27 tampilan block yang terantai

Gambar 3.27 adalah tampilan setelah proses *chaining* bekerja pada web dan data tersebut sudah terantai dari satu data ke data yang lain atau yang bisa disebut data *blockchain*.



Gambar 3.28 Tampilan data voter yang dienkripsi. Tampilan pada gambar 3.28 memperlihatkan data dari voter yang sudah dienkripsi dan terantai berurutan, data ini berisi data *voter* (Nama, Alamat, Nomer KTP), data yang dipilih *voter* dan timestamp untuk menunjukkan kapan *voter* melakukan pemilihan dan hash dari ketiga data tersebut. Dan data ini berada dalam *block*.

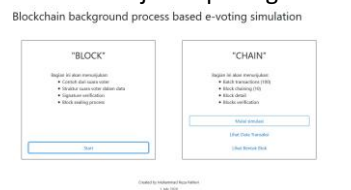
3. HASIL DAN PEMBAHASAN

Tampilan Background Menu Utama

Pada tahap tampilan ini, aplikasi hanya untuk pengembangan penelitian, aplikasi berikut menunjukkan bagaimana proses yang terjadi pada aplikasi utama tahap demi tahap.

Tampilan background manu utama ini menampilkan fitur yang sedikit berbeda karena hanya untuk melihat bagaimana data dibentuk berupa *Block*, dan merantai data yang dibentuk sehingga terciptalah proses *Blockchain*.

Menu utama dari aplikasi ini menunjukkan bagaimana membuat *block* dari sebuah data dan merantainya. Tampilan dari background menu utamaditunjukkan pada gambar 4.21 dibawah.



Gambar 4.21 Menu Utama Background Proses

Pembuatan Block Data

Pada tampilan ini, proses pembuatan data dilakukan, data yang dibuat *block* disini adalah data dari *voter* yang telah memilih kandidat namun dalam bentuk enkripsi, id disini adalah data *voter* yang dienkripsi RSA, data voter-nya yaitu nomer ktp, nama, alamat, dan password.

Private key disini adalah hasil dari generating antara data *voter* tadi dan kandidat yang dipilih dan mengasilkan *private key*, proses ini dilakukan komputer sebagai syarat pembuatan *block* data. Tampilan dari pembuatan *block* data ditunjukkan pada gambar 4.22 dibawah.



Gambar 4.22 Pembuatan Block Data

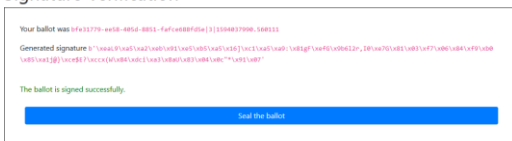
Pada simulasi ini halaman ini, *private key* berada pada sistem setting milik *framework* django. *Private key* ini berfungsi untuk *hash* tambahan yang dibuat oleh *framework* ini yang berfungsi untuk membuat *hash* baru dan membuat *signature* atau tanda terima.

Private key ini akan saling berkesinambungan dengan *public key* yang berada pada *framework* ini. Karena untuk melakukan enkripsi pada suatu data diperlukan *private key*, dalam hal ini menggunakan metode RSA.

Sebaliknya *public key* disini berfungsi untuk mendekrip data yang dienkripsi sebelumnya dengan menggunakan *private key*. Dengan adanya *public key* ini, sistem tidak perlu repot mencari *key* baru yang akan memakan waktu yang sangat lama, ditambah dengan jumlah data dari para *voter* yang bisa sangat banyak jika pemilihan dilakukan secara besar.

Signature Verification

Disini setelah data diproses, data akan di-verifikasi apakah data yang sudah dimasukan tadi telah benar atau tidak, dan dalam proses ini kode tanda tangan akan dibuat oleh sistem, data ini berupa kode yang akan diberikan kepada *voter* untuk menunjukkan bahwa *voter* telah memilih kandidat. Tampilan pada gambar 4.23 dibawah.



Gambar 4.23 Signature Verification

Block Detail

Pada tampilan ini, data dari *voter* sudah menjadi sebuah *block*, isi dari data *block* ini adalah *previous hash*, atau *hash* dari data yang sebelumnya, jika data tidak terantai maka hanya akan keluar data *genesis*, *transaction hash* adalah enkripsi data dari *voter* yang sudah memilih, data tersebut berisi nomer ktp, nama, alamat, *password*, dan status. *Block hash* disini adalah enkripsi dari data *voter* yang sudah memilih, *transaction hash*, *nonce*, dan *timestamp*. *Nonce* disini adalah data hasil kerja dari sebuah sistem *blockchain* untuk menunjukkan

bahwa suatu *block* data sudah dibuat atau disebut juga *proof of work*.

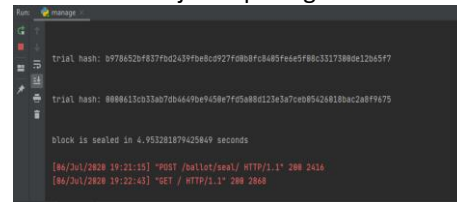
Timestamp disini adalah data berupa waktu yang dibuat oleh sistem, waktu disini diambil pada saat sistem membuat *block*, bukan pada saat *voter* melakukan pemilihan. Tampilan dari *block detail* ditunjukkan pada gambar 4.24 dibawah.



Gambar 4.24 Block Detail

Tampilan Pycharm Command Block

Tampilan ini untuk memperjelas bahwa sistem benar-benar melakukan pembuatan *block* dengan benar. Pada saat aplikasi dijalankan dan melaksanakan perintah untuk membuat *block*, maka tampilan dari *command line* aplikasi akan tetap terlihat. Tampilan pada saat aplikasi selesai membuat *block*, tampilan pada *command line* juga berhasil membuat *block*, dengan kata lain data sebuah *block* sudah dibuat. Tampilan *command line block* ditunjukkan pada gambar 4.25 dibawah.



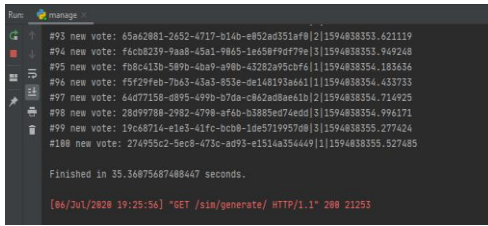
Gambar 4.25 Command line block

Proses Chaining

Pada proses ini akan menunjukkan bagaimana proses *chaining* atau merantai data satu dengan data seterusnya, data yang dirantai berbentuk enkripsi RSA dari data *voter* yang sudah memilih kandidat dan *timestamp* dari kapan proses data ini dibuat.

Karena proses ini terjadi spontan ketika seorang *voter* melakukan pemilihan lalu menekan tombol pilih dan sistem melakukan penyimpanan *database*, maka proses pembuatan data tidak terlihat diinterface.

Namun dalam proses ini ketika dimenu utama sebelah kanan pada tab "CHAINING" dan setelah menekan tombol start maka proses *chaining* dimulai. Tampilan dari proses *chaining* dilakukan ditunjukkan pada gambar 4.26dibawah

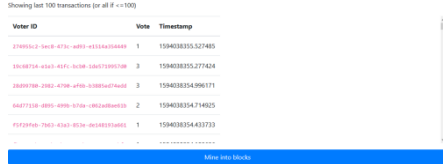


Gambar 4.26 Proses Chaining

Data Enkripsi Voter

Setelah proses chaining selesai, maka data siap dibuat menjadi block-block data sesuai dari pengembang ingin membuat berapa block dari data voter yang sudah valid. Data tersebut seperti nomer ktp, nama, alamat, nomer kandidat yang dipilih dan timestamp. Tampilan data enkripsi voter ditunjukkan pada gambar 4.27 dibawah.

List of unconfirmed votes

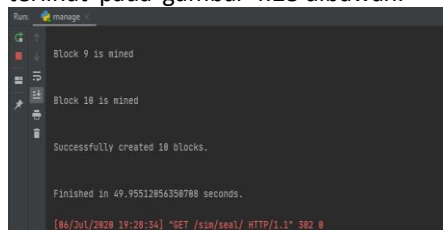


Gambar 4.27 Data Enkripsi Voter

Proses Mining Block

Proses ini adalah proses paling penting dari aplikasi ini yaitu proses mining block yang sudah dirantai atau bisa disebut inilah inti dari proses dari teknologi Blockchain bekerja.

Data dari para voter yang sudah dienkripsi tadi dibuatkan block-block data, block data ini juga akan dirantai dengan block data seterusnya yang juga disebut proses chaining tadi. Tampilan pada web hanya akan meload data, sedangkan proses pada command line atau sistem akan terlihat pada gambar 4.28 dibawah.



Gambar 4.28 Proses Mining Block

List Of Block

Setelah block dibuat, tampilan akan menunjukkan daftar dari block-block yang dibuat tersebut. Data dari block tersebut adalah berupa tabel yang berisi previous hash adalah enkripsi SHA3-256 dari sebuah block, lalu ada block hash atau isi dari block tersebut yang dienkripsi, lalu merkle root hash atau enkripsi dari mekanisme block ini dibuat, nonce atau proof of work dari setiap block dan timestamp dari kapan block ini dibuat.

Tampilan dari list of block tersebut ditunjukkan pada gambar 4.29 dibawah.

List of blocks



Gambar 4.29 List Of Block

Proses Verifikasi Block

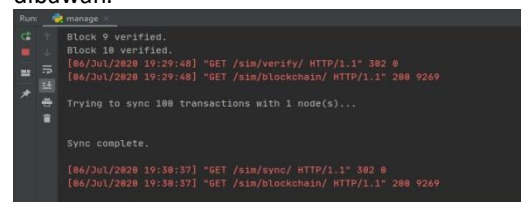
Proses ini untuk verifikasi apakah block data yang dibuat tadi datanya benar dan tidak berubah. Proses ini akan mengecek rantai data dari hash block satu dengan block seterusnya agar memastikan semua block data terantai dan tidak bisa diubah disalah satu block. Tampilan dari verifikasi block berupa command ditunjukkan pada gambar 4.30 dibawah.



Gambar 4.30 Proses Verifikasi Block

Proses Sinkron Block

Proses mensinkronkan data ini adalah untuk memastikan bahwa data masih sama dan belum berubah. Proses ini sangat krusial jika tidak dilakukan data pada block akan berantakan satu dengan seterusnya karena tidak sinkron. Tampilan command line ditunjukkan pada gambar 4.31 dibawah.



Gambar 4.31 Proses Sinkron Block

List Of Confirmed Votes

Ini adalah tampilan data dari para voter yang sudah memilih tadi dan sudah dibentuk dalam bentuk block data. Pada tampilan ini menunjukkan data berupa Voter ID adalah data voter yang sudah memilih kandidat yang berbentuk enkripsi, lalu nomer kandidat yang dipilih, timestamp atau waktu pembuatan data, dan hash dari data-data sebelumnya.

Pada halaman ini juga menampilkan berapa banyak suara voter yang didapatkan oleh

kandidat-kandidat yang tersedia. Data ini background dari halaman hasil sementara pada aplikasi utama di website *e-vote*. Tampilan dari *list of confirmed votes* ditunjukkan pada gambar 4.32 dibawah.

List of confirmed votes

#	Voter ID	Vote	Timestamp	Hash	Block
1	413124f1-17c1-4a78-af71-f120872207a9	2	2020-07-06 19:25:20	111773a6a76244a6c1...	go to block
2	774b6394-203a-4853-8832-ae4353a32704	1	2020-07-06 19:25:20	7782f8e112f7893a6d...	go to block
3	8d739488-323a-4813-8d81-89626a484854	1	2020-07-06 19:25:21	7d77a797327f11ca6d...	go to block
4	8b6d4584-7439-4ac1-f1-4810-40d8a7f51202	2	2020-07-06 19:25:21	4a6867fa2207068036...	go to block
5	13d72918-4071-487f-8788-9822a16a7174	2	2020-07-06 19:25:21	4a67f8a6a0879a2961...	go to block

Gambar 4.32 List Of Confirmed Votes

Blockchain Data Detail

Pada halaman ini, data dari sebuah block yang sudah terantai akan terlihat, datayang ditampilkannya yaitu:

1. *Previous hash*, atau enkripsi dari data sebelumnya, jika data merupakan data pertama maka hanya akan terlihat angka nol sepanjang angka enkripsi block.
2. *Merkle root hash* atau enkripsi dari mekanisme sebuah block dibuat.
3. *Block hash* atau enkripsi dari data voter.
4. *Nonce* atau proof of work dari block.
5. *Timestamp* atau waktu pada saat block dibuat.
6. Dan *confirmation* atau *block* yang dikonfirmasi dari *block* sebelumnya, jika *block* sebelumnya ada 5 *block*, maka *confirmation* hanya akan muncul 4 karena *block* yang sedang dilihat sudah di konfirmasi.

Tampilan dari *blockchain* data detail ini ditunjukkan pada gambar 4.33 dibawah.

Block #1

Previous hash	00
Merkle root hash	00
Block hash	00
Name	27851
Timestamp	2020-07-06 19:27:46.38 (waktu lokal)
Confirmations	0 confirmations

Gambar 4.33 Blockchain Data Detail

Data Transaksi

Transactions

#	Voter ID	Cast	Timestamp	Hash
1	413124f1-17c1-4a78-af71-f120872207a9	2	2020-07-06 19:25:20	111773a6a76244a6c1...
2	774b6394-203a-4853-8832-ae4353a32704	1	2020-07-06 19:25:20	7782f8e112f7893a6d...
3	8d739488-323a-4813-8d81-89626a484854	1	2020-07-06 19:25:21	7d77a797327f11ca6d...
4	8b6d4584-7439-4ac1-f1-4810-40d8a7f51202	2	2020-07-06 19:25:21	4a6867fa2207068036...
5	13d72918-4071-487f-8788-9822a16a7174	2	2020-07-06 19:25:21	4a67f8a6a0879a2961...

Tampilan data transaksi ini adalah isi dari jumlah voter yang ada dalam *block*, transaksi hash bisa juga disebut mekanisme dari sebuah *block* dibuat dalam bentuk enkripsi atau disebut juga dengan *merkle root hash*. Tampilan ditunjukkan pada gambar 4.34 dibawah.

Gambar 4.34 Data Transaksi

Data Detail 9 Confirmation

Pada halaman ini, data dari sebuah *block* yang sudah terantai akan dengan 9 konfirmasi dari data sebelumnya, data yang ditampilkan yaitu:

1. *Previous hash*, atau enkripsi dari data sebelumnya, jika data merupakan data pertama maka hanya akan terlihat angka nol sepanjang angka enkripsi *block*.
2. *Merkle root hash* atau enkripsi dari mekanisme sebuah *block* dibuat.
3. *Block hash* atau enkripsi dari data voter.
4. *Nonce* atau *proof of work* dari block.
5. *Timestamp* atau waktu pada saat *block* dibuat.

Dan *confirmation* atau *block* yang dikonfirmasi dari *block* sebelumnya, jika *block* sebelumnya ada 9 *block*, maka *confirmation* hanya akan muncul 8 karena *block* yang sedang dilihat sudah di konfirmasi.

Tampilan dari *blockchain* data detail ini ditunjukkan pada gambar 4.35 dibawah.

Block #2

Previous hash	00
Merkle root hash	00
Block hash	00
Name	27851
Timestamp	2020-07-06 19:27:46.38 (waktu lokal)
Confirmations	0 confirmations

Gambar 4.35 Data 9 Confirmation

Data Detail 1 Confirmation

Pada halaman ini, data dari sebuah *block* yang sudah terantai akan dengan 9 konfirmasi dari data sebelumnya, data yang ditampilkan yaitu:

1. *Previous hash*, atau enkripsi dari data sebelumnya, jika data merupakan data pertama maka hanya akan terlihat angka nol sepanjang angka enkripsi *block*.
2. *Merkle root hash* atau enkripsi dari mekanisme sebuah *block* dibuat.
3. *Block hash* atau enkripsi dari data voter.
4. *Nonce* atau *proof of work* dari block.
5. *Timestamp* atau waktu pada saat *block* dibuat.

Dan *confirmation* atau *block* yang dikonfirmasi dari *block* sebelumnya, jika *block* sebelumnya ada 9 *block*, maka *confirmation* hanya akan muncul 8 karena *block* yang sedang dilihat sudah di konfirmasi.

Tampilan dari *blockchain* data detail ini ditunjukkan pada gambar 4.36dibawah.



Gambar 4.36 Data 1 Confirmation

Validasi Data

Setelah semua *voter* melakukan *voting*, mereka akan mendapat kode dimana kode itu adalah hasil transaksi dari proses voting yang mereka lakukan. Kode ini juga berfungsi sebagai data validasi apakah *voter* sudah melakukan voting atau belum, apakah *voter* memilih kandidat sesuai dengan pilihannya atau belum, dan apakah voter memilih kandidat lebih dari satu atau tidak.

Kode ini dapat dicek oleh admin atau petugas untuk pengecekan data dan validasi dengan cara melihat kedalam databasenya, pengawas akan mensinkronkan data *voter* yang sudah memilih dengan data transaksi yang ada.

Gambar dari kode transaksi disajikan dalam gambar 4.37 sebagai berikut.

ID	id_pencalon	id_pemilih	tanda_terima
3	12	8	3BZUNWXCFCY
4	13	9	4CATLJQDCHK
5	12	1	5NVRALQJPU
6	14	2	6UEQJWZNXG
7	14	3	7HVSMCIYPW
8	14	4	8DERLOTYZN
9	14	10	9ULAIPXBHD
10	14	13	10ZCSUBTFD

Gambar 4.37 Data Tanda Terima

4. SIMPULAN

Berdasarkan hasil implementasi dan pengujian yang dilakukan terhadap sistem yang dibuat maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi website e-vote memudahkan dalam membuat pemilihan umum secara online
2. Teknologi BlockChain yang diterapkan membuat aplikasi menjadi lebih aman
3. Teknologi BlockChain ini dapat digunakan pada aplikasi website electronic voting
4. Aplikasi e-vote membuat pengembangan teknologi BlockChain ini menjadi lebih baik dari segi penerapan

5. DAFTAR PUSTAKA

- [1] Yusrianto Toisutta, Eka. 2013. Pengaksesan Daun Secara Random Pada Hash Tree. Jurnal, Bandung. Institut Teknologi Bandung
- [2] Ayed, A. B. 2017. A Conceptual Secure Blockchain-Based Electronic Voting System. International Journal of Network Security & Its Applications, 9(3): 1-9. doi:10.5121/ijnsa.2017.9301
- [3] Liu, Y., & Wang, Q. 2017. An E-voting Protocol Based on Blockchain. IACR Cryptology ePrint Archive, 2017, 1043.
- [4] Maya Kadarina, Trie. 2019. Pengenalan Bahasa Pemrograman Python Menggunakan Aplikasi Games Untuk Siswa/I di Wilayah Kembangan Utara. Jurnal, Jakarta. Universitas Mercu Buana Jakarta
- [5] Yoan Rizaldi, Dimas. 2019. Implementasi Multichain sebagai Alternatif Solusi Keamanan dan Privasi Data pada Komunikasi Perangkat Pintar Rumah. Jurnal, Surabaya. Universitas Negeri Surabaya
- [6] Setiadi, Angga. Harihayati, Tati. 2016. PENERAPAN SQLITE PADA APLIKASI PENGATURAN WAKTU UJIAN DAN PRESENTASI. Jurnal, Jakarta. Universitas Komputer Indonesia
- [7] d Azwanti, N. 2017. Perancangan E-Voting berbasis Web. Jurnal Komputer Terapan, 3(2): 119-132.
- [8] Noizat, P. 2015. Chapter 22 -Blockchain Electronic Vote. Chuen, D. L.K. (Ed.) Handbook of Digital Currency (pp 453-461). San Diego: Academic Press.
- [9] Miller.P , "The cryptocurrency enigma," in Digital Forensics: Threatscape and Best Practices, no. August 2015, Elsevier Inc., 2015, pp. 1–25.