ANALISIS DAN IMPLEMENTASI KONTROL AKSES PADA WEB BERBASIS BLOCKCHAIN

Muhammad Rifki Fauzan[1], Parman Sukarno [2], Aulia Arif Wardana [3]

[1,2,3] Universitas Telkom, Bandung

mrifkifauzan@students.telkomuniversity.ac.id[1], parmansukarno@telkomuniversity.ac.id[2], auliawardan@telkomuniversity.ac.id[3]

Abstrak

Kontrol akses merupakan suatu komponen yang sangat krusial dan penting dalam keamanan sistem. Biasa digunakan untuk melindungi suatu data penting atau sebuah tindakan yang sensitif. Salah satunya adalah penggunaan pada *web application*. Pada kontrol akses terdapat dua komponen, yaitu otentikasi dan otorisasi. Otentikasi digunakan untuk metode pengamanan dalam memverifikasi pengguna. Lalu, Otorisasi digunakan untuk mengatur hak akses pada web application dengan tujuan membatasi akses penggunanya. Penyimpanan hak akses pada *web* banyak menggunakan *database*. Namun, terdapat sebuah resiko apabila menggunakan penyimpanan pada *database*, yaitu serangan *SQL Injection*. Untuk mengatasi masalah tersebut terdapat teknologi baru, yaitu *blockchain*, *blockchain* secara singkat merupakan *database* terdistribusi. Keunggulan dari *blockchain* salah satunya adalah dapat mencegah serangan *SQL Injection*. *Blockchain* tidak menggunakan perintah seperti *SQL* dan menggunakan teknik *hash* untuk keamanan penyimpanan data. Pada penelitian ini penyimpanan hak akses akan digantikan dengan teknologi *blockchain*. Oleh karena itu, pembuatan *web application* yang akan dibangun adalah pengaturan kontrol akses menggunakan *blockchain*.

Kata kunci : kontrol akses, *blockchain, web application*

Abstract

Access control is a crucial and important component in system security. Commonly used to protect important data or a sensitive action. One of them is the use on the web application. In access control there are two components, namely authentication and authorization. Authentication is used for security methods in verifying users. Then, Authorization is used to set permissions on the web application for the purpose of restricting the access of its users. Storage permissions on the web use many databases. However, there is a risk of using storage on the database, which is a SQL Injection attack. To solve the problem there is a new technology, namely blockchain, blockchain is briefly a distributed database. The advantage of blockchain one of them is that it can prevent SQL Injection attacks. Blockchains do not use SQL-like commands and use hash techniques for data storage security. In this study the storage of access rights will be replaced with blockchain technology. Therefore, the creation of a web application to be built is the setting of access control using blockchain.

Keywords: access control, blockchain, web application

1. Pendahuluan

Latar Belakang

Pada era digital, teknologi informasi berkembang semakin pesat serta sebagian besar sudah mempengaruhi kebutuhan hidup manusia. Salah satunya pada perkembangan *web*. Melalui *web*, semua orang dapat mengakses

banyak informasi-informasi yang tersebar luas diinternet [1]. Namun, tidak semua informasi yang ada di *web* dapat diakses oleh semua orang. Ada hal yang dinamakan kontrol akses, yang digunakan untuk mengatur hak akses seseorang untuk dapat mengakses suatu informasi tertentu. Pada kontrol akses terdapat dua komponen, yaitu otentikasi dan otorisasi. Otentikasi digunakan untuk metode pengamanan dalam memverifikasi pengguna [12]. Otorisasi merupakan sebuah komponen keamanan yang digunakan untuk melindungi suatu data penting atau sebuah tindakan yang sensitif. Otorisasi dapat digunakan untuk memberikan hak istimewa kepada seseorang, agar dapat mengakses suatu data yang memiliki hak akses terbatas [2].

Otorisasi ini dimaksudkan agar suatu data hanya dapat terlindungi dengan membatasi hak akses. Dalam aplikasi *web*, otorisasi dimulai dengan memberikan hak akses untuk user sesuai kebutuhan. Pengaturan hak akses biasa digunakan dengan menandai suatu akun *user* dengan menyimpan pemberian kewenangannya (*role*) di *database* [3]. Penggunaan *database* sebagai tempat menyimpan pengaturan hak akses ini masih sering digunakan. Namun, terdapat resiko yang cukup tinggi apabila menggunakan sebuah *database*, salah satunya adalah serangan *SQL Injection*. Serangan *SQL Injection* ini adalah suatu serangan yang memasukan atau menyuntikan perintah *SQL* melalui input data yang terdapat pada aplikasi *web* [4]. Serangan *SQL Injection* dapat melakukan berbagai perintah *SQL* yang memungkinkan penyerang dapat mengambil data-data yang ada pada database. Apabila serangan ini dapat dilakukan, tidak hanya dapat mencuri data saja, bahkan penyerang dapat melakukan perubahan sampai penghapusan data [5]. Terdapat sebuah teknologi bernama *blockchain* yang merupakan *database* terdistribusi yang akan digunakan sebagai database atau tempat menyimpan hak akses tersebut. *Blockchain* memiliki sebuah kelebihan yang dapat menghindari serangan *SQL Injection* [6], hal ini dikarenakan pengunaan *Blockchain* tidak seperti perintah *SQL* dan juga teknik penyimpanan pada *Blockchain* menggunakan teknik *hash* untuk membentuk rantai satu blok dengan blok yang lainnya dimana pada blok tersebut terdapat data yang disimpan [13].

Oleh karena itu, kontrol akses yang dilakukan pada penelitian kali ini adalah dengan menggunakan *blockchain*. Model kontrol akses menggunakan RBAC (*Role Base Access Control*) karena hak akses yang diberikan sesuai dengan tingkatan jabatan aktor. Pada penelitian ini, penulis menambahkan suatu fitur keamanan, yaitu pengaturan kontrol akses pada *web* berbasis *blockchain*. Hal ini dimaksudkan sebagai pengamanan *web* untuk mencegah *user* mengakses halaman yang tidak diizinkan atau hanya dapat melakukan sesuatu pada halaman yang diizinkan saja pada suatu *web*. Harapannya dengan adanya penelitian ini dapat meningkatkan keamanan aplikasi *web*.

Topik dan Batasannya

Berdasarkan pemaparan latar belakang, dibuat perumusan masalah yang akan menjadi acuan untuk penelitian. Permasalahan yang disebutkan pada latar belakang, yaitu masih terdapatnya kelemahan pada pengimplementasian kontrol akses yang menggunakan penyimpanan dengan *database* biasa, sehingga rentan terhadap peretasan data yang ada pada *database*.

Batasan masalah dari penelitian ini, aplikasi *web* yang dibuat adalah *web* aplikasi pengamanan ijazah dan transkrip berbasis *blockchain* dan *smart contract*.

Tujuan

Berdasarkan perumusan masalah yang telah ditentukan, maka dibuat juga tujuan dari penelitian ini, yaitu membangun aplikasi *web* dengan pengaturan kontrol akses berbasis *blockchain*.

Organisasi Tulisan

Dalam bab 2, menjelaskan penelitian yang relevan yang digunakan untuk mendukung penelitian. Bab 3 menjelaskan desain sistem yang dibangun untuk penelitian. Bab 4 menjelaskan pengujian dan hasilnya, serta membahas penelitian yang dilakukan. Bab 5 menjelaskan hasil penelitian.

2. Studi Terkait

2.1 Blockchain

Secara singkat *blockchain* merupakan basis data terdistribusi dari *record* atau *public ledger* dari semua transaksi atau peristiwa digital yang sudah dieksekusi dan lalu dibagikan pada pihak yang berpartisipasi. Setiap transaksi yang dilakukan diverifikasi dengan *consensus* mayoritas oleh partisipan yang ada pada sistem. Setelah dimasukan ke dalam sistem, data tidak akan pernah bisa dihapus. Transaksi yang dilakukan akan direkam dalam satu buah *block*, dimana setiap *block* ini terhubung dengan *block* lain yang ada sebelum atau sesudah sebuah transaksi dilakukan. *Blockchain* berisikan *record* tertentu dan dapat diverifikasi dari semua transaksi yang pernah dilakukan. *Bitcoin* merupakan mata uang berbentuk digital yang *peer-to-peer* terdesentralisasi. *Bitcoin* adalah contoh paling popular yang menggunakan teknologi blockchain. Mata uang *Bitcoin* dapat digunakan dalam perdagangan digital yang fungsinya sama dengan mata uang pada dunia nyata [7] [8].

2.2 Ethereum

Ethereum merupakan sebuah teknologi yang ada pada *blockchain* yang menyediakan sebuah aplikasi *platform* terdesentralisasi untuk operasi *smart contract* [9]. *Ethereum* juga memungkinkan penggunanya untuk membuat aplikasi terdesentralisasi dan *smart contract* sendiri, bahkan dapat membuat aturan semaunya untuk kepemilikan sendiri, format transaksi, dan fungsi-fungsi transisinya [10]. Dalam *platform Ethereum* ini, *Bitcoin* berfungsi sebagai mata uang digital yang dapat digunakan untuk membelanjakan atau mentransfer asset digital pada *platform*.

2.3 Kontrol Akses

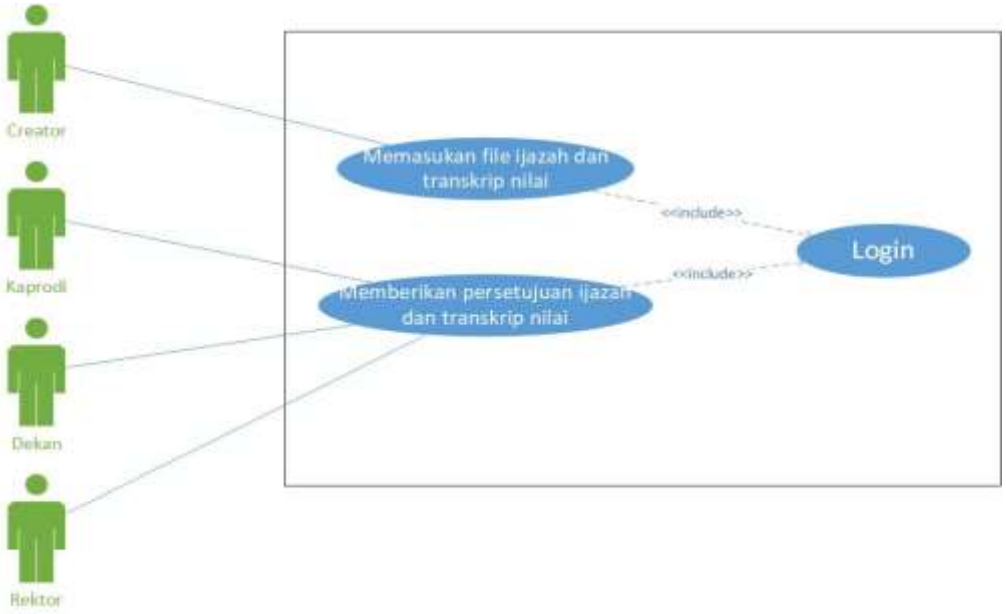
Kontrol akses merupakan proses dimana suatu user diberikan akses dan hak untuk melihat sistem, atau informasi yang ada. Kontrol akses memiliki dua komponen, yaitu otentikasi dan otorisasi. Otentikasi digunakan untuk metode pengamanan dalam memverifikasi pengguna. Biasanya otentikasi adalah kombinasi dari e-*mail* dan *password* untuk dapat mengakses sebuah akun [12]. Pengunaan otentikasi biasanya digunakan pada aplikasi *web* untuk memverifikasi penggunanya, otentikasi yang digunakan berbasis pada *password*. Jika *e-mail* dan *password* benar, maka sistem memverifikasi bahwa *user* dapat mengakses informasi yang ada. Namun, ada pula sistem otentikasi yang menggunakan dua faktor. Seperti saat *e-mail* dan *password* telah dinyatakan benar, terdapat satu sistem keamanan lagi, dimana akan meminta sebuah kode rahasia yang hanya dikirimkan ke pengguna yang berhak.

Otorisasi merupakan sebuah komponen keamanan yang digunakan untuk melindungi suatu data penting atau sebuah tindakan yang sensitif. Otorisasi adalah cara untuk mengatur tingkatan hak akses terhadap suatu data. Otorisasi dapat digunakan untuk memberikan hak istimewa kepada seseorang, agar dapat mengakses suatu data yang memiliki hak akses terbatas. Otorisasi juga sangat krusial pada komponen keamanan suatu sistem. Hal ini juga untuk menghindari kebocoran data yang diakibatkan oleh faktor *insider threat*, dimana hanya orang yang berwenang saja yang dapat mengakses data penting [11].

3. Sistem yang Dibangun

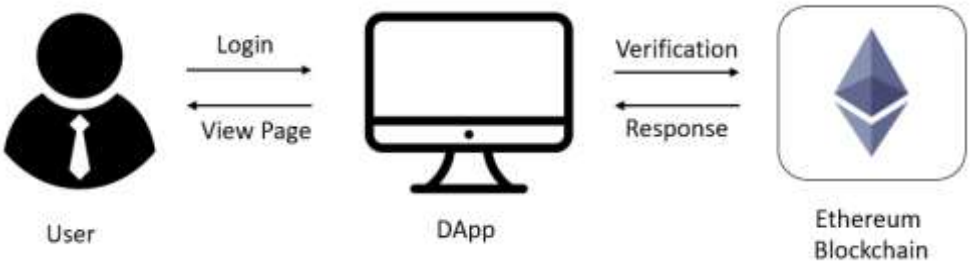
3.1 Perancangan Sistem

Aplikasi *web* yang akan dibangun akan memiliki empat aktor, yaitu creator, ketua program studi (kaprodi), dekan, dan rektor. *Creator* memiliki akses untuk membuat ijazah dan transkrip nilai pada *web*. Kaprodi memiliki akses untuk memberikan persetujuan kepada ijazah dan transkrip nilai pada *web*. Dekan memiliki akses untuk memberikan persetujuan setelah kaprodi menyetujui ijazah dan transkrip nilai. Rektor juga memiliki akses untuk memberikan persetujuan setelah dekan menyetujui ijazah dan transkrip nilai.



Gambar 1 *Use Case* Diagram Aplikasi *Web*

Pengaturan hak akses akan menggunakan *Ethereum Blockchain* sebagai *database* atau media penyimpanannya. Aplikasi web yang dibangun menggunakan *Distributed Application* (DApp) dengan bantuan *web3js*. Dengan menggunakan *web3js* ini memungkinkan untuk dapat berinteraksi dengan *Ethereum Blockchain*. Pada saat *user* melakukan *login* sistem akan melakukan pengecekan terhadap *hash* data *user* pada *block* yang ada pada *Ethereum*. Apabila *hash* terverifikasi, maka *user* akan diarahkan ke halaman yang sesuai dengan pengaturan hak akses sebelumnya.



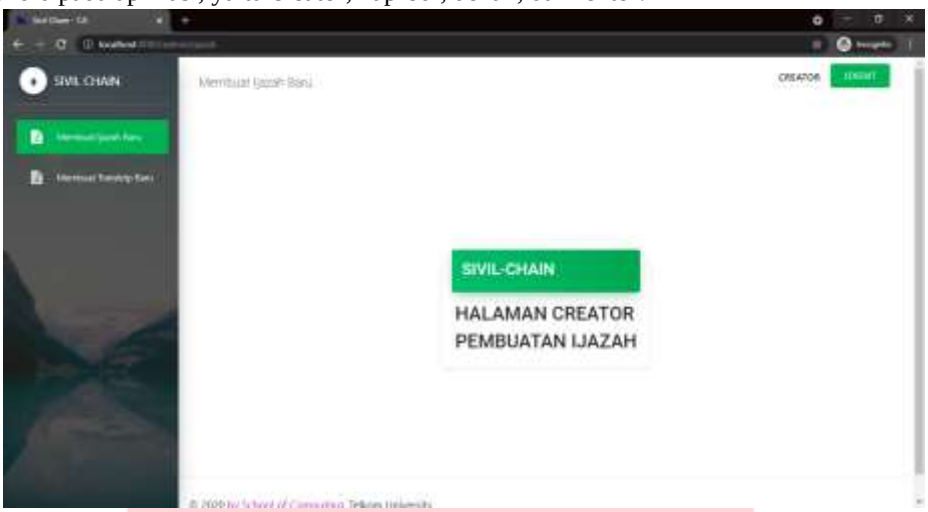
Gambar 2 Gambaran Sistem Aplikasi *Web* yang Akan Dibangun

3.2 Proses Pengujian

Proses pengujian adalah dengan melakukan serangan keamanan kepada aplikasi *web* berbasis *blockchain* yang dibangun. Serangan yang dilakukan adalah untuk menguji tingkat keamanan dari aplikasi web dalam segi integritas dari data kontrol akses yang ada pada blockchain, apabila serangan dapat dilakukan, maka data pengguna yang ada pada blockchain dapat dilakukan perubahan kewenangannya. Serangan akan dilakukan menggunakan serangan *SQL Injection* pada aplikasi *web*. Pengujian dilakukan dengan melakukan serangan *SQL Injection* pada *form login* dengan *Union-based SQLI*. Serangan *SQL Injection* dengan menggunakan sintaks *query* terhadap *blockchain*. Sintaks *query SQL Injection* akan digunakan pada *form login* yang ada pada aplikasi *web* yang bertujuan untuk menguji apakah dengan meninjeksi *query* dapat terlihat isi dari *database blockchain*. Apabila serangan *SQL Injection* tidak menghasilkan sesuatu dari isi *blockchain*, maka terbukti bahwa dengan menggunakan *blockchain* sebagai *database* dapat mencegah serangan *SQL Injection* pada aplikasi *web*.

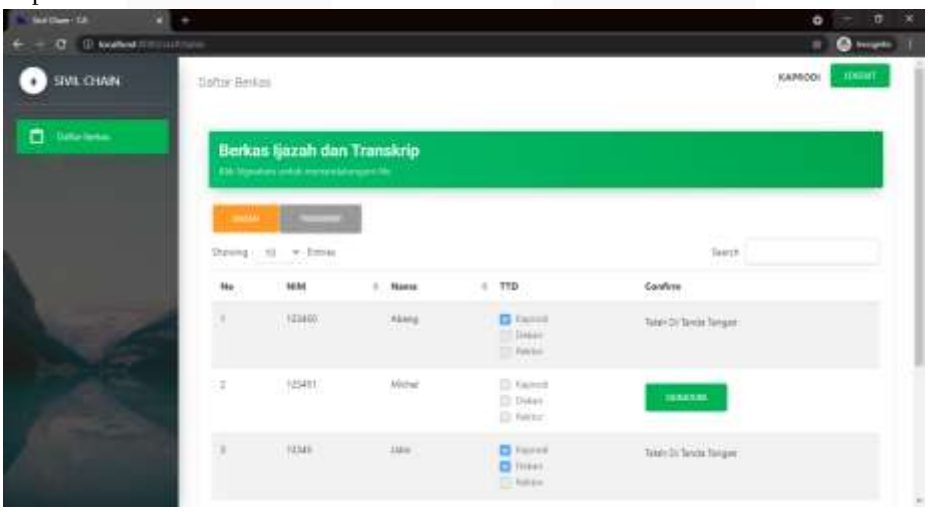
4. Evaluasi

Aplikasi *web* yang dibangun dengan menggunakan *blockchain* sebagai *database* untuk penyimpanan data beserta *role* setiap akun. Aplikasi dibangun dengan *NodeJs* sebagai *Back-End* dan *VueJs* sebagai *Front-End*. Terdapat empat *role* pada aplikasi, yaitu creator, kaprodi, dekan, dan rektor.

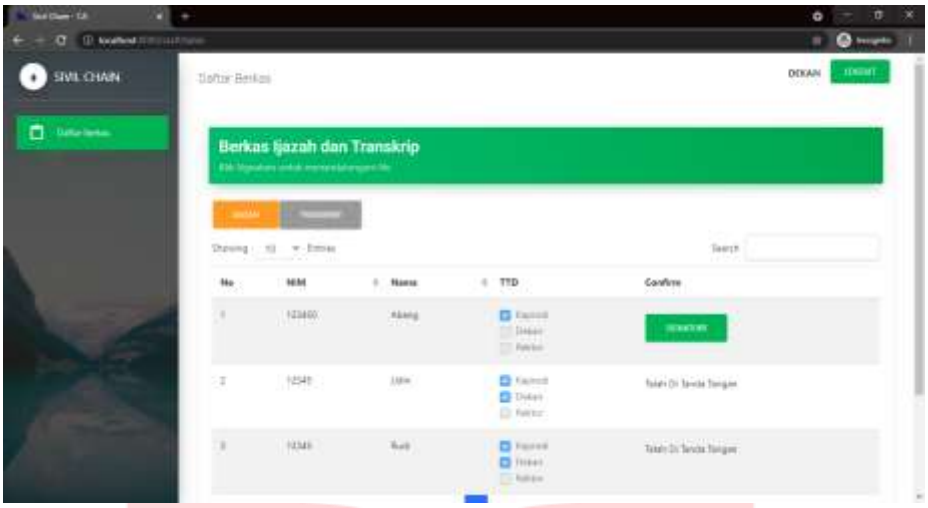


Gambar 3 Halaman Creator

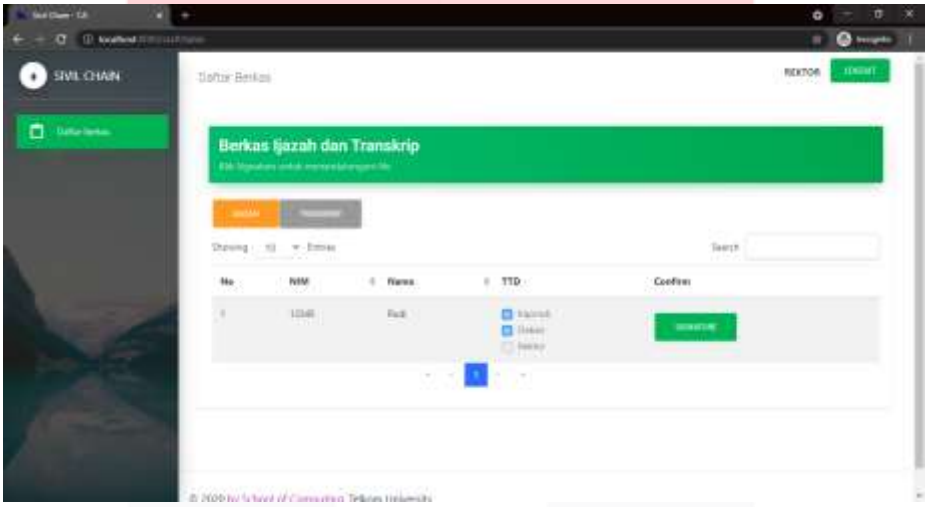
Pada *role* creator, memiliki hak untuk membuat ijazah dan transkrip nilai mahasiswa. Lalu pada *role* kaprodi, dekan, dan rektor memiliki hak untuk menyetujui penerbitan ijazah dan transkrip nilai secara berurut berdasarkan tingkatan jabatan. Pada halaman kaprodi (Gambar 4) akan menampilkan daftar ijazah dan transkrip nilai yang perlu disetujui. Apabila kaprodi telah memberikan persetujuan pada ijazah atau transkrip nilai, maka akan tertampil pada halaman dekan (Gambar 5). Lalu apabila dekan telah memberikan persetujuan pada ijazah atau transkrip nilai, maka akan tertampil pada halaman rektor (Gambar 6) untuk disetujui dan sebagai persetujuan akhir dari ijazah dan transkrip nilai.



Gambar 4 Halaman Kaprodi



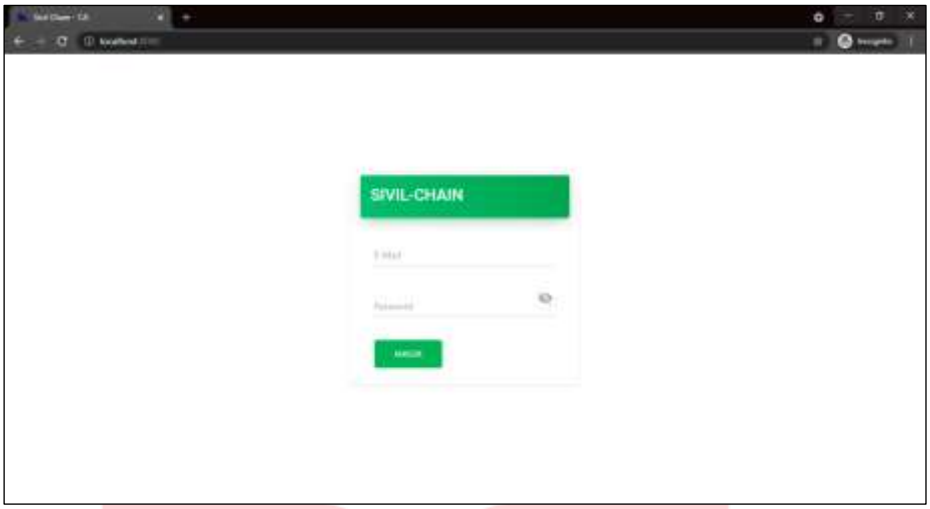
Gambar 5 Halaman Dekan



Gambar 6 halaman Rektor

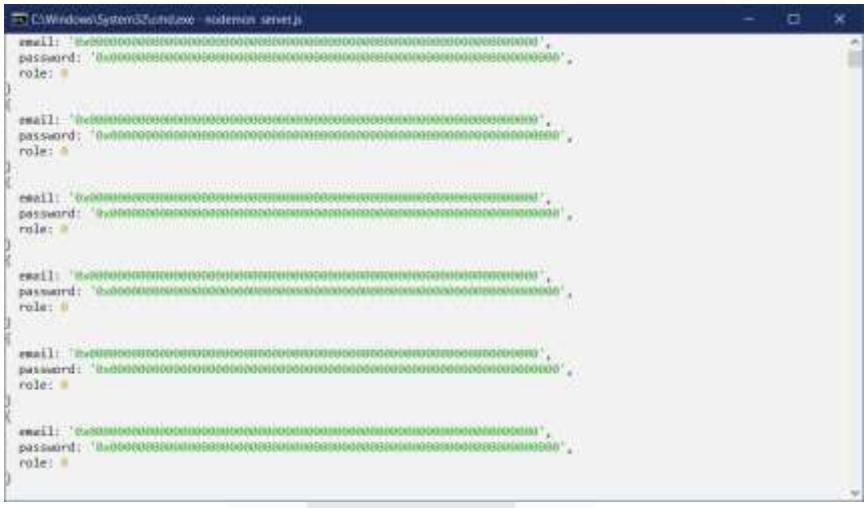
4.1 Hasil Pengujian

Pengujian dilakukan dengan menggunakan serangan *SQL Injection* dengan metode *Union-based SQLI*. Alasan menggunakan *Union-based SQLI* sebagai serangan *SQL Injection* karena merupakan salah satu dari serangan injeksi yang umum digunakan. Serangan ditujukan pada halaman *login* pada aplikasi yang dibangun, untuk membuktikan bahwa dengan penggunaan *blockchain* sebagai *database* dapat mencegah serangan *SQL Injection* sehingga dapat menjaga keamanan data beserta hak akses yang terdapat di dalam *database*.



Gambar 7 Halaman Login

Serangan menggunakan *wordlist* yang diambil dari [15]. Penulis mengambil sebagian dari serangan *Union-based SQLI*, yaitu sebanyak 150 serangan *SQL Injection*. Serangan pada mengguakan metode POST karena fungsi halaman *login* yang dibangun menggunakan metode POST. Hasil dari pecobaan 150 serangan yang dilakukan 100% berhasil dicegah oleh *blockchain* sebagai *database*. Tidak ada serangan yang menyebabkan *error* pada saat serangan dilakukan terhadap aplikasi. Aplikasi juga sengaja tidak menggunakan teknik *code* pada *NodeJs* yang dapat mencegah serangan *SQL Injection.*



Gambar 8 Hasil serangan *SQL Injection*

Pada Gambar 8 merupakan keluaran yang dihasilkan pada saat melakukan serangan *SQL Injection* pada aplikasi, hal ini membuktikan bahwa penggunaan *blockchain* sebagai *database* terbukti aman dari *SQL Injection*.

4.2 Analisis Hasil Pengujian

Berdasarkan pengujian yang dilakukan pada aplikasi *web* yang diimplementasikan *blockchain* sebagai *database* terbukti bahwa dapat mencegah serangan *SQL Injection*. Hal tersebut dikarenakan cara yang dilakukan oleh *blockchain* untuk menyimpan datanya berbeda dengan *database* pada umumnya. *Database* yang biasa digunakan menggunakan *SQL* untuk memasukan datanya ke dalam *database*-nya, sedangkan *blockchain* menggunakan metode lain, yaitu menggunakan *solidity* sebuah pemrograman berorientasi objek yang digunakan

untuk membuat *smart contract* yang berisikan *code* fungsi yang dibuat sesuai kebutuhan. Teknik penyimpanan pada *blockchain* menggunakan teknik hash untuk membentuk rantai satu blok dengan blok yang lainnya dimana pada blok tersebut terdapat data yang disimpan. Sehingga dapat disimpulkan untuk penggunaan *blockchain* sebagai *database* aman untuk digunakan.

5. Kesimpulan

Sistem pada aplikasi *web* dibangun untuk membuat kontrol akses dengan menggunakan *blockchain* sebagai *database*. Kontrol akses yang diimplementasikan pada aplikasi *web* berhasil dilakukan dengan pengaturan *role* sebanyak empat aktor, diantaranya *creator*, kaprodi, dekan, dan rektor. Setiap *role* memiliki hak aksesnya masing-masing, *creator* memiliki hak untuk membuat ijazah dan transkrip nilai mahasiswa. Sedangkan kaprodi, dekan, dan rektor memiliki hak akses untuk persetujuan penerbitan ijazah dan transkrip nilai mahasiswa. Persetujuan dilakukan secara berurutan berdasarkan jabatan yang dimulai dari kaprodi, dekan, dan terakhir oleh rektor.

Penggunaan *blokchain* bertujuan untuk mencegah terjadinya serangan terhadap *database*, yaitu serangan *SQL Injection* yang dapat merugikan apabila data dapat terlihat oleh orang yang tidak memiliki wewenang untuk mengaksesnya. Berdasarkan pengujian yang telah dilakukan, *blockchain* terbukti dapat mencegah serangan *SQL Injection* sehingga aman untuk digunakan sebagai *database*. Hal ini dikarenakan metode *blockchain* sangat berbeda dengan *database* pada umumnya, yaitu tidak menggunakan *SQL* untuk memasukan datanya ke dalam *database*. Sehingga pada pengujian semua serangan yang dilakukan berhasil dicegah dan hal ini membuktikan bahwa *blockchain* dapet mencegah serangan *SQL Injection.*

Referensi

[1] Prasetiadi, A. E. (2020). Web 3.0: Teknologi Web Masa Depan. *Jurnal Industri Elektro dan Penerbangan*, *1*(3).

[2] Andersen, M. P., Kolb, J., Chen, K., Fierro, G., Culler, D. E., & Popa, R. A. (2017). Wave: A decentralized authorization system for iot via blockchain smart contracts. *University of California at Berkeley, Tech. Rep*.

[3] Ru, R., Selo, S., & Widyawan, W. (2017). Implementasi *Role-Based Access Control* (RBAC) pada Pemanfaatan Data Kependudukan Ditingkat Kabupaten. *Prosiding Semnastek*.

[4] OWASP. "SQL Injection." owasp.org. https://owasp.org/www-community/attacks/SQL_Injection (Diakses Oktober 5, 2020).

[5] Riadi, I., Umar, R., & Sukarno, W. (2018). Vulnerability of Injection Attacks Against The Application Security of Framework Based Bebsites Open Web Access Security Project (OWASP). *J. Inform*, *12*(2), 53-57.

[6] Yunus, M. A. M., Brohan, M. Z., Nawi, N. M., Surin, E. S. M., Najib, N. A. M., & Liang, C. W. (2018). Review of SQL Injection: Problems and Prevention. *JOIV: International Journal on Informatics Visualization*, *2*(3-2), 215-219.

[7] Crosby, M. (2015). Blockchain Technology Beyond Bitcoin. [online] Available at: https://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf [Accessed 7 Oct. 2020].

[8] Carlozo, L. (2017). What is blockchain?. *Journal of Accountancy*, *224*(1), 29.

[9] Gencer, A. E., Basu, S., Eyal, I., Van Renesse, R., & Sirer, E. G. (2018, February). Decentralization in bitcoin and ethereum networks. In *International Conference on Financial Cryptography and Data Security* (pp. 439-457). Springer, Berlin, Heidelberg.

[10] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *white paper*, *3*(37).

[11] Claycomb, W. R., Huth, C. L., Flynn, L., McIntire, D. M., Lewellen, T. B., & Center, C. I. T. (2012). Chronological Examination of Insider Threat Sabotage: Preliminary Observations. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *3*(4), 4-20.

[12] Aini, Q., Rahardja, U., Madiistriyatno, H., & Setiaji, Y. D. M. (2018). Pengamanan Pengelolaan Hak Akses Web Berbasis *Yii Framework. Syntax: Jurnal Informatika*, *7*(1), 52-63.

[13] Noorsanti, R. C., Yulianton, H., & Hadiono, K. (2018). Blockchain-Teknologi Mata Uang Kripto (*Crypto Currency*).

[14] Putri, M. C. I., Sukarno, P., & Wardana, A. A. (2020). *Two factor authentication framework based on ethereum blockchain with dApp as token generation system instead of third-party on web application.* Register: Jurnal Ilmiah Teknologi Sistem Informasi, 6(2), 74-85.

[15] Tasdelen, Ismail. (2019). SQL Injection Payload List. [online] Available at: https://ismailtasdelen.medium.com/sql-injection-payload-list-b97656cfd66b [Accessed 13 May. 2021].