JISKa, Vol. 6, No. 3, SEPTEMBER, 2021, Pp. 139 – 148
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Optimasi Keamanan Web Server terhadap *Serangan Broken Authentication* Menggunakan Teknologi *Blockchain*

Imam Riadi [(1)], Herman [(2)], Aulyah Zakilah Ifani [(3)*]
[1] Sistem Informasi, Fakultas Sains dan Teknologi Terapan, Universitas Ahmad Dahlan, Yogyakarta
[2,3] Teknik Informatika, Fakultas Sains dan Teknologi Terapan, Universitas Ahmad Dahlan, Yogyakarta
e-mail : imam.riadi@is.uad.ac.id, hermankaha@mti.uad.ac.id, aulyah1908048022@webmail.uad.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 29 April 2021, direvisi 25 Juni 2021, diterima 11 Juli 2021, dan dipublikasikan 22 September 2021.

Abstract

The aspect of the internet that needs to be considered a security is the login system. The login system usually uses a username and password as an authentication method because it is easy to implement. However, data in the form of usernames and passwords are very vulnerable to theft, so it is necessary to increase the security of the login system. The purpose of this research is to investigate the security of the system. Whether the system is good at protecting user data or not, minimizing execution errors from the system and minimizing risk errors on the system so that the login system can be used safely. This research is conducted to test the system security with Burp Suite *on the login system that has been built. Testing the security of this system by experimenting with POST data which is secured using blockchain technology makes the data sent in the form of hash blocks safer and more confidential so that the system is safer than before. Blockchain technology has successfully secured usernames and passwords from broken authentication attacks. By using the* Burp Suite *testing system, login is more specific in conducting security testing.*

Keywords: Authentication, Broken Authentication, Blockchain, Burp Suite, Login System

Abstrak

Salah satu aspek di internet yang perlu diperhatikan keamanannya adalah sistem *login*. Sistem *login* biasanya menggunakan *username* dan *password* sebagai metode autentikasi karena mudah dalam mengimplementasikannya. Data *username* dan *password* sangat rentan diretas sehingga perlu dilakukan peningkatan keamanan pada sistem *login*. Penelitian bertujuan untuk mengetahui keamanan dari sistem dalam melindungi data pengguna, meminimalkan kesalahan eksekusi dari sistem serta mengurangi risiko *error* pada sistem, sehingga sistem *login* bisa digunakan secara aman. Penelitian ini dilakukan untuk menguji keamanan sistem dengan Burp Suite pada sistem *login* yang dibangun. Pengujian keamanan sistem ini dengan percobaan data POST yang diamankan menggunakan teknologi *blockchain* membuat data yang dikirimkan dalam bentuk blok *hash* menjadi lebih aman dan rahasia sehingga sistem lebih aman daripada sebelumnya. Teknologi *blockchain* berhasil mengamankan *username* dan *password* dari serangan *broken authentication*. Pengujian menggunakan Burp Suite pada sistem *login* lebih spesifik dalam melakukan pengujian keamanan.

Kata Kunci: Autentikasi, Broken Autenthication, Burp Suite, Sistem *Login*, Teknologi *Blockchain*

1. PENDAHULUAN

Keamanan terhadap data, informasi, dan sistem secara keseluruhan semakin penting seiring berkembangnya teknologi informasi. Adanya sistem *login* merupakan salah satu aspek di internet yang perlu diperhatikan keamanannya (Ramadhan & Ariyani, 2018). Kebutuhan informasi dalam internet yang luas memberikan kemudahan dalam mengaksesnya. Data atau informasi menjadi sangat rentan terhadap pencurian sehingga perlu menjaga integritas data ataupun informasi. Dalam mengimplementasikan suatu web diperlukan proses *login*. Sistem keamanan dan proses

login biasanya menggunakan *username* dan *password* sebagai metode autentikasi. Hal ini digunakan karena kemudahan dalam mengimplementasikannya (Sudiarto Raharjo et al., 2017). Akan tetapi autentikasi menggunakan *username* dan *password* rentan terhadap peretasan. Terutama ketika *username* dan *password* disimpan dalam sebuah *database*. Hal ini terbukti ketika dilakukan *vulnerability assesment* menggunakan beberapa *tools* seperti *openvas,* Burp Suite dan *wireshark*. Kelemahan lainnya, ketika sistem autentikasi diretas, sulit bagi pengguna untuk mengetahuinya (A. W. P. Putra et al., 2018).

Autentikasi adalah suatu pembuktian identitas terhadap suatu entitas seperti pada mesin, kartu kredit, dan orang (Rusdan & Sabar, 2020). Autentikasi dibagi menjadi tiga kategori di antaranya yaitu: *What the entity knows* contoh berupa kata sandi, kedua *What the entity owns* seperti kartu pintar, kunci privasi atau tiket kerberos, dan ketiga *What the entity* yang mencakup teknik autentikasi berdasarkan fitur *biometric* pengguna (sidik jari, bentuk wajah, bentuk tangan, dll.) (Sudiarto Raharjo et al., 2017). Salah satu tekologi inovasi yang mampu menyelesaikan permasalahan tersebut adalah teknologi *Blockchain.*

Blockchain merupakan sebuah teknologi dalam pertukaran informasi tanpa melibatkan pihak ketiga. Informasi berupa informasi dalam bentuk digital, entri data transaksi, aset (Dilley et al., 2016). *Blockchain* adalah teknologi dengan *database* terdistribusi disimpan dan dibagikan ke pengguna yang berwewenang (Parizi et al., 2018) (Bouscaren, 1989). Paling tidak *blockchain* melibatkan tiga unsur teknologi yang sebetulnya sudah ada sejak lama, yaitu internet, protokol dari perangkat lunak, dan kriptografi (Fadlil et al., 2020). Teknologi *blockchain* membuat peretas akan sulit mengubah dan memodifikasi data yang sama di semua komputer di saat yang sama karena membutuhkan waktu yang sangat lama untuk memecahkan kode enkripsi pada setiap blok data di seluruh jaringan komputer (Riadi, Umar, & Busthomi, 2020).

Di balik teknologi *blockchain* terdapat 6 karakteristik utama yaitu *blockchain* adalah kriptografi yang di dalamnya tercatat proses enkripsi yang tingkat keamanan setiap transaksi tinggi. *Blockchain* adalah akuntansi yang di dalamnya tercatat tentang sebuah transaksi. *Blockchain* adalah rantai, ini dikarenakan *blockchain* terdiri dari kumpulan blok, di mana blok sebelumnya harus sama sehingga dapat menyambung seperti sebuah rantai. *Blockchain* merupakan catatan terdistribusi di mana data transaksi dalam *blockchain* tersimpan pada suatu buku besar terdistribusi (distribusi *ledger*) di seluruh *nodes* dan sulit dimanipulasi oleh *adversaries*. *Blockchain* adalah *mining* karna setiap seseorang berhasil melakukan validasi kebenaran transaksi maka akan mendapatkan sejumlah imbalan dalam bentuk *native coin* dari *blockchain* tersebut. *Blockchain* adalah *smart contract* karna selain menyimpan data dan transaksi, *blockchain* juga bisa mengeksekusi kontrak perjanjian yang telah disimpan sebelumnya (Rahardja et al., 2019).

Jenis *record* yang terdapat pada sistem *blockchain* yaitu blok dan transaksi. Setiap transaksi *blockchain* tersimpan dalam satu blok secara bersama. Setiap blok membentuk jaringan yang berisi algoritma kriptografi. Algoritma kriptografi digunakan untuk mengambil data dari blok sebelumnya dan diubah ke *Compact String* yang dapat mendeteksi sabotase (Fauzan, 2018). Setiap blok memiliki nilai *hash* yang didapatkan dari blok sebelumnya (Hu et al., 2019). Cara kerja fungsi *hash* yaitu panjang variabel diubah ke dalam bentuk biner. Setiap biner memiliki panjang yang sama. Fungsi *hash* digunakan dalam sistem keamanan, salah satu contohnya autentikasi pesan, penyimpanan *password*, dan tanda tangan digital (H. F. Putra et al., 2019). Untuk mengetahui seberapa besar tingkat keamanan dari suatu sistem penting dilakukan *penetration testing,* berupa simulasi terhadap serangan nyata yang mungkin menyerang sistem tersebut.

Penetration testing merupakan mengamati serangan dan menganalisis risiko terkait pelanggaran dari keamanan. Penguji tidak hanya dapat mengetahui keberadaan celah bagi *hacker,* tetapi juga dapat mengeksploitasi lebih jauh untuk mengevaluasi tingkat kerentanan sebuah sistem (Azis & Fattah, 2019). *Penetration testing* memerlukan analisa intensif pada kerentanan sistem yang di dapatkan dari kelemahan sistem. Seluruh data analisa yang telah dilakukan akan didokumentasikan dan serahkan ke *user* yang pemilik sistem juga dampak beserta solusi yang didapatkan penguji dari celah keamanan yang ada (Pangalila et al., 2015). Penelitian ini

menggunakan Burp Suite untuk menguji keamanan sistem. Burp Suite merupakan *tool* untuk melakukan keamanan *open source* yang digunakan untuk menjalankan dan menguji fitur keamanan pada sebuah aplikasi *website* (Sai Kiran et al., 2020). Burp Suite digunakan untuk menangkap aliran data dengan mengatur sebagai pendengar *proxy* yang bertindak sebagai *server proxy HTTP* lokal (Joshi & Kumar, 2016). Burp Suite secara keseluruhan merupakan kerangka pengujian untuk penetrasi web. Burp Suite terbagi menjadi dua yaitu ada *community edition* dan *profesional edition.* Selain itu Burp Suite sebagai platform terintegrasi berbasis Java untuk melakukan pengujian keamanan suatu aplikasi web. Burp Suite awalnya hanya merupakan aplikasi *proxy server* untuk melakukan *intercept* baik terhadap http-request ataupun http-response ke server dan web *application* (T & Sasikala, 2019). Terdapat beberapa kerentanan yang menjadi risiko pada sebuah aplikasi web yaitu injeksi*, insecure direct object references, broken authentication and session management (XSS), security misconfiguration, sensitive data exposure, cross site request forgery (CSRF), cross site scripting (XSS), unvalidated redirects and forwards, using components with known vulnerabilities, missing function level access control* (Guntoro et al., 2020).

Kebanyakan dari pengguna menggunakan pengujian *vulnerability* untuk meningkatkan kesadaran tentang pentingnya keamanan informasi. Kerentanan (*vunlnerability)* dari sebuah sistem disebabkan oleh faktor eksternal dan faktor internal (Wibowo et al., 2019). Dalam menguji *vulnerability* sistem dapat dilakukan dengan 2 tipe yaitu *external testing* dan *internal testing. External Testing* merupakan analisa terhadap informasi *public* yang tersedia. Untuk menampilkan jumlah *network acces point* merupakan *internal testing* yang mewakili beberapa *logical* dan *phisical segment* (Harjowinoto et al., 2016)*.* metode dalam *vulnerability* yaitu pertama, *passive vulnerability testing* dengan melakukan pengujian terhadap kontrol *login,* konfigurasinya, dan kontrol *web application* sehingga dapat memetakan target sistem. kedua, *active vulnerability testing,* di mana pengujian dilakukan dengan memanipulasi input terhadap kerentanan yang ada, pengambilan hak akses. Ketiga, *aggressive vulnerability testing,* dilakukan *reverse engineering* terhadap *software* dan *system* (Harjowinoto et al., 2016).

Kerentanan (*vulnerability*) banyak diterapkan dan dilakukan contohnya pengujian terhadap web OJS versi 3.0, hasil pengujian didapatkan celah untuk melakukan serangan *Cross-site Scripting* (XSS) (Riadi, Yudhana, et al., 2020). Penelitian juga dilakukan pada sistem administrasi rumah sakit X (Harjowinoto et al., 2016), *security auditing* pada *vulnerable machine* (Sitinjak et al., 2020), dan juga pada pengujian celah keamanan pada CMS (Kunang et al., 2013). Selain menggunakan *OpenVAS* penelitian juga dilakukan pada aplikasi *smart payment* menggunakan *OWASP,* pengujian menggunakan *framework OWASP* menguji kerentanan terhadap serangan XSS (*Cross-site Scripting*) (Riadi, Umar, & Lestari, 2020)*.* Ada beberapa jenis *vulnerability* di antaranya pemindaian berbasis jaringan, pemindaian berbasis *host*, pemindaian jaringan nirkabel, pemindaian aplikasi. Pemindaian berbasis jaringan untuk mengidentifikasi keamanan jaringan. Pemindaian berbasis *host* untuk mengidentifikasi kerentanan server, *host* jaringan lainnya. Pemindaian jaringan nirkabel lebih berfokus pada titik serangan dalam nirkabel. Pemindaian aplikasi untuk mendeteksi kerentanan perangkat lunak. Pemindaian basis data mengidentifikasi titik lemah dalam basis data (Laksmiati, 2020). Dalam penelitian ini menggunakan *broken authentication* sebagai serangan akan digunakan.

Broken authentication merupakan kerentanan web yang terjadi karna kesalahan konfigurasi manajemen *session*. Hal yang harus diperhatikan dari *broken autentication* yaitu pertama, *password strength*, di mana pada aplikasi yang yang kita bangun harus memiliki level minimal dari keamanan *password*, yang dapat dilihat pada panjang *password* dan kompleksitasnya. Kedua, *password use*, di mana aplikasi yang kita buat harus mempunyai batasan *user* mengaksesnya dalam tenggang waktu tertentu. Ketiga, *password storage*, di mana *password* yang kita miliki tidak boleh disimpan dalam aplikasi, dalam hal ini *password* harus ada dalam keadaan terenkripsi. Keempat, *issue* lainnya yang berhubungan contohnya *password* di dalam *source code* tidak boleh dalam bentuk *hard-coded.* Kelima, *Session ID Protection* hal ini digunakan biasanya untuk server mengidentifikasi *user* yang akan masuk ke dalam *session* menggunakan *session ID* (Hassan et al., 2018). *Broken authentication* memiliki fungsi untuk autentikasi dan manajemen *session* yang tidak dapat diterapkan dengan baik, memungkinkan

penyerang menyusup untuk mendapatkan *username* dan *password* dan mengeksploitasi kelemahan implementasi untuk mengasumsikan identitas pengguna lain secara permanen atau sementara (OWASP, 2017).

Gambar 1. Skenario Broken *Authentication*.

Gambar 1 menunjukkan *attacker* mengirimkan permintaan kredensial pengguna yang dihasilkan sampai sistem menemukan itu benar. Server memverifikasi kredensial pengguna dan membuat sesi yang kemudian disimpan dalam *database*. Setelah kredensial yang didapat ditebak cocok dengan *database*, sistem mengirimkan respons ke *attacker* atau penyerang dengan akses di akun.

Teknologi *blockchain* digunakan dalam penelitian ini untuk mengamankan data *username* dan *password* pengguna. Sebagaimana diketahui *blockchain* menggunakan fitur tanda tangan digital untuk melakukan transaksi sehingga data dari *user* tidak dapat diubah atau dirusak, hal ini menjadi kelebihan dari *blockchain*. Selain itu, *blockchain* menggunakan sistem terdesentralisasi yang dapat membuat transaksi lebih aman, cepat, dan lancar (Parizi et al., 2018). Sehingga penelitian ini menggunakan *blockchain* sebagai tujuan untuk meningkatkan keamanan *username* dan *password* dari suatu sistem *login*. Pengujian sistem menggunakan *tool* Burp Suite untuk memastikan sistem *login* aman digunakan.

2. METODE PENELITIAN

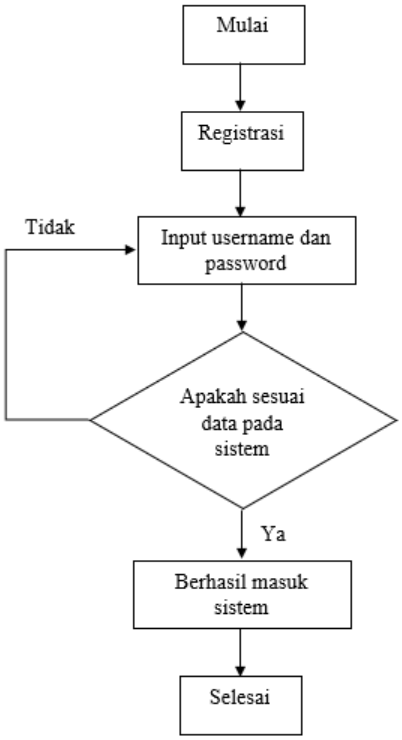
2.1. Objek Penelitian

Sistem *login* adalah aplikasi yang digunakan sebagai objek pada penelitian ini. Aplikasi ini merupakan aplikasi atau *website* yang akan digunakan di berbagai aplikasi lainnya. Sistem *login* terdapat *username* dan *password* yang akan diisikan terlebih dahulu sebelum masuk ke sistemnya. Memasukkan *username* dan *password* menjadi sangat rentan terhadap peretasan. Bahnyaknya *tools* yang tersedia membuat data pengguna menjadi terancam. Hal ini tentunya akan sangat berbahaya apabila dibiarkan terus menerus. Oleh karena itu pada penelitian ini akan dilakukan optimasi keamanan. Dengan menggunakan teknologi *blockchain* diharapkan dapat digunakan sebagai pengamanan sistem *login*. Pengujian dalam penelitian ini menggunakan serangan *broken authentication.*

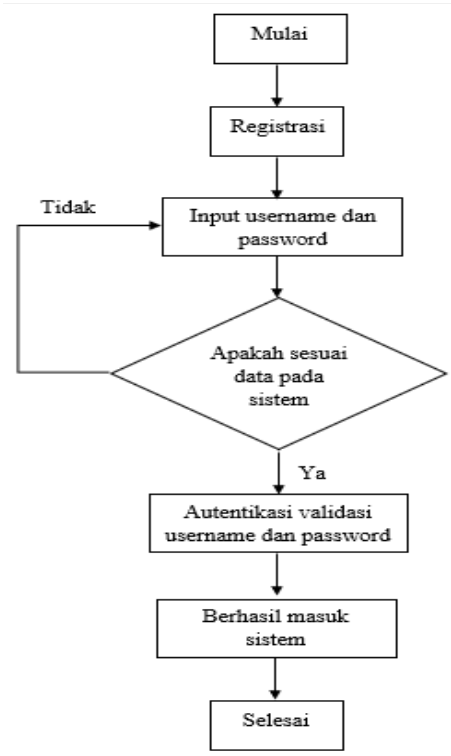
2.2. Desain Sistem

Desain *flowchart* sistem menggambarkan alur kerja sistem yang akan dirancang. Sistem mempunyai kemampuan untuk melakukan pengecekan data yang ada pada sistem *login*. Sistem mempunyai kemampuan untuk melakukan autentikasi validasi *username* dan *password*. Sebelum melakukan autentikasi sistem terlebih dahulu melakukan pengecekan data yang ada pada sistem *blockchain*. Tahapan *flowchart* sistem, pertama pengguna atau *user* akan melakukan *login* terlebih dahulu. Setelah *login* pengguna memasukan *username* dan *password*. Sistem akan mengecek apakah data ada pada sistem *login*, jika data yang dimasukkan sudah benar maka sistem akan melakukan autentikasi validasi *username* dan *password*, setelah semua

proses selesai maka pengguna akan masuk sistem. Berikut perancangan *flowchart* sistem dapat dilihat pada Gambar 2.



Gambar 2. *Flowchart* Sistem Sebelum Menggunakan *Blockchain*.



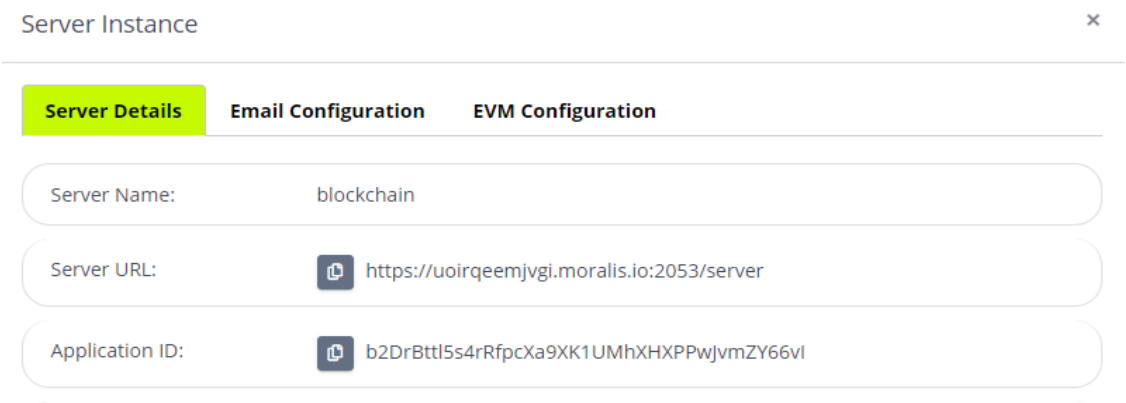
Gambar 3. *Flowchart* Sistem Setelah Menggunakan *Blockchain*.

Gambar 2 dan Gambar 3 menunjukkan *flowchart* sistem di mana pengguna atau *user* akan melakukan *login* terlebih dahulu sebelum masuk ke sistem. Setelah memasukan *username* dan *password* sistem akan melakukan autentikasi *username* dan *password*. Jika sudah sesuai maka proses *login* akan berhasil. Pada Gambar 2 setelah pengguna melakukan registrasi lalu *login* ke sistem, sistem akan mengecek dan berhasil masuk apabila data yang dimasukkan sesuai. Berbeda dengan *flowchart* yang ada pada gambar 3 yang menggunakan pengecekan autentikasi *username* dan *password* terlebih dahulu sebelum masuk ke dalam sistem, di mana Gambar 3 terdapat proses autentikasi dengan MetaMask. MetaMask digunakan sebagai jembatan antara sistem *login* dengan *blockchain* Ethereum. Sedangkan Gambar 2 tidak melalui proses tersebut sehingga sangat rentan terhadap peretasan.

3. HASIL DAN PEMBAHASAN

Sistem *login* pada penelitian ini memanfaatkan penggunaan *platform blockchain* Ethereum yang mengimplementasikan teknologi *blockchain* dan *smart contract*. Dengan web3.js sebagai *application programming interface* (API) untuk menghubungkan *browser* dengan ekstensi yang dinamakan MetaMask sebagai jembatan antara sistem *login* dengan *blockchain* Ethereum. MetaMask ini bertindak sebagai wallet Ethereum untuk pengelolaan informasi. Sementara itu untuk *smart contract* dibangun dengan menggunakan tools Visual Studio Code (VSCode). Pada sistem *login* juga menggunakan Moralis sebagai autentikasi ke MetaMask. Moralis mengambil informasi dari akun MetaMask. Untuk menghubungkan Moralis dengan MetaMask perlu adanya penyinkronan yang dilakukan di VSCode.

User harus menjadi anggota jaringan *blockchain* ketika jadi anggota maka *user* akan mempunyai *user* ID Ethereum. Sebuah aplikasi berbasis *blockchain* Ethereum harus menjalankan *smart contract*. Sehingga untuk *smart contract* akan ditanda tangani terlebih dahulu oleh orang-orang yang sudah mempunyai ID Ethereum untuk menjalankan sebuah aplikasi berbasis Ethereum. *User* tersebut selanjutnya jika ingin *login* maka harus sudah terdaftar sebagai penandatangan *smart contract*. Berikut merupakan tampilan dari akun Moralis pada Gambar 4.



Gambar 4. Server URL dan *application* ID akun Moralis.

Gambar 4 merupakan server URL dan *application* ID pada akun Moralis yang akan disalin untuk menyinkronkan ke MetaMask. Link URL tersebut kemudian disalin dan dimasukkan ke dalam *source code* di VSCode. Berikut Gambar 5 merupakan URL untuk menyinkronkan akun MetaMask dan Moralis.

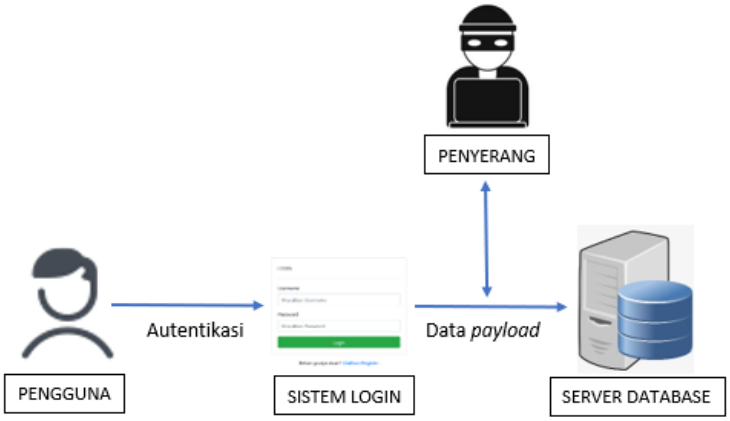
```
<script>
    Moralis.initialize("b2DrBttl5s4rRfpcXa9XK1UMhXHXPPwJvmZY66vI"); // Application id from moralis.io
    Moralis.serverURL = "https://uoirqeemjvgi.moralis.io:2053/server"; //Server url from moralis.io
```

Gambar 5. *Source code* penyinkronan akun MetaMask dan Moralis.

Gambar 5 menunjukkan bagaimana menghubungkan antara akun MetaMask dengan Moralis. Menghubungkan akun tersebut terlebih dahulu masuk ke akun Moralis poc.moralis.io untuk mendapatkan *application* ID dan server URL tersebut. *Application ID from Moralis.io* digunakan untuk menghubungkan web3. Sedangkan server URL digunakan untuk menghubungkan Moralis. Ketika sistem sudah terhubung selanjutnya mengisikan *username* dan *password* dan akan mendapatkan notifikasi dari MetaMask dan diminta untuk memberikan tanda tangan terlebih dahulu.

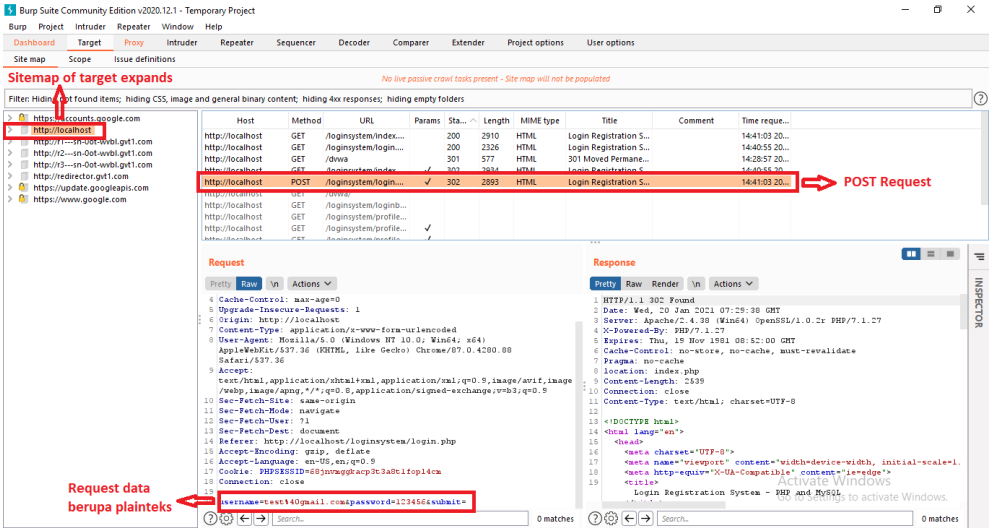
Hasil penerapan teknologi *blockchain* sebagai autentikasi pada sistem *login* mempunyai kemampuan untuk melakukan pengecekan data yang ada pada sistem *blockchain.* Komponen-komponen yang ada pada sistem meliputi menu registrasi, menu *login*, menu lihat profil, menu *update* profil dan *logout.* Halaman registrasi merupakan gambaran awal ketika pengguna belum memiliki akun untuk *login* yang digunakan untuk mendaftarkan *username* dan *password* sebelum masuk ke sistem. Menu *login* merupakan langkah pertama sebelum masuk ke sistem. Halaman ini dibuat untuk memberikan batasan kepada pihak yang tidak berkepentingan agar tidak dapat mengakses dan mengoleh data tanpa melakukan *login* terlebih dahulu. Sehingga sebelum melakukan *login*, pengguna atau *user* harus melakukan registrasi terlebih dahulu untuk mendapatkan akun. *User* memasukkan *username* dan *password* dengan benar maka akan diarahkan ke menu awal dari sistem. Setelah *login* berhasil maka *user* dapat melihat profil dan melakukan *update* data dari *user*.

Pengujian dilakukan menggunakan aplikasi Burp Suite sebagai *software* untuk melakukan percobaan serangan *broken authentication* pada sistem *login*. Autentikasi yang diperlukan yaitu *username* dan *password* yang sudah terdaftar sebelumnya. Autentikasi dengan *username* dan *password* sangat mudah untuk mengimplementasikannya. Akan tetapi, menggunakan *username* dan *password* membuat para peretas mudah untuk melakukan serangan. Gambar 6 akan dipaparkan skenario yang akan dilakukan sebelum menggunakan teknologi *blockchain*.



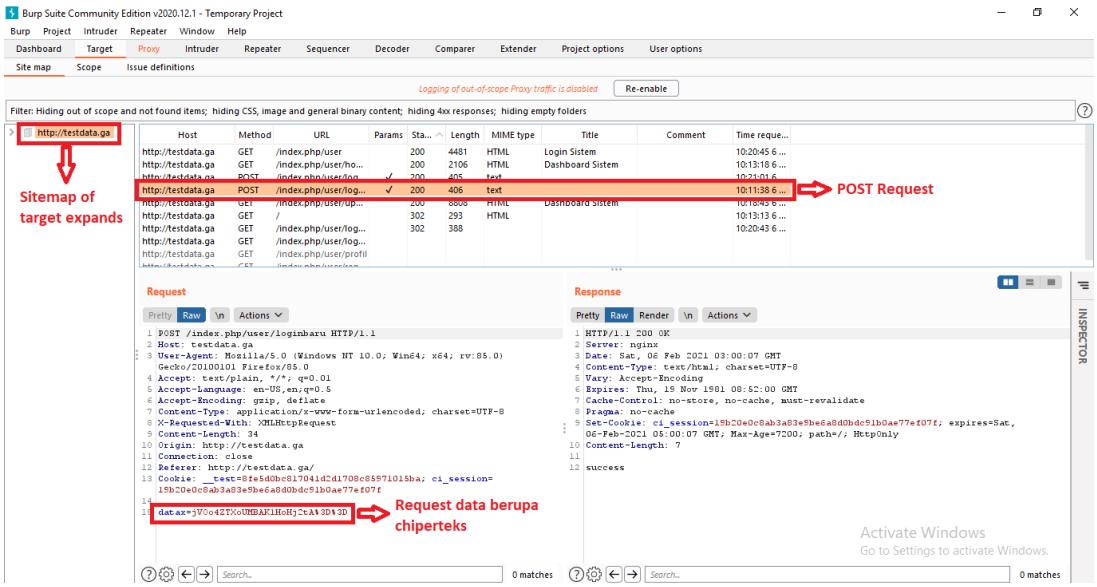
Gambar 6. Skenario keamanan sistem *login*.

Gambar 6 menunjukkan mengenai skenario keamanan yang ada pada sistem. Pengguna akan meginputkan *username* dan *password* ke dalam sistem *login*. Data *payload* akan dikirimkan ke *database* server. Penambahan *blockchain* pada autentikasi sistem *login* membuat penyerang tidak dapat membaca isi *payload* data yang dikirimkan pengguna. Pengujian sistem menggunakan *tool* Burp Suite sebagai *software* untuk melakukan penetrasi pada halaman *login* untuk mencoba masuk ke dalam sistem. Penetrasi ini melakukan bantuan beberapa aplikasi dalam pengujiannya nanti. Pengujian ini difokuskan pada celah keamanan *login* dan autentikasi pada sistem. Percobaan data POST yang diamankan dengan teknologi *blockchain* membuat data yang dikirimkan dalam bentuk blok *hash.* Blok *hash* menjadi lebih aman dan rahasia, data POST dikirimkan ke *database* server. Pengujian sebelum mengunakan *blcokchain* dapat dilihat pada Gambar 7.



Gambar 7. Aplikasi Burp Suite menampilkan *username* dan *password*.

Gambar 7 menunjukkan hasil pengujian setelah diimplementasikannya *blockchain* pada sistem *login*. Aplikasi Burp Suite menampilkan *username* dan *password* ketika masuk ke dalam sistem. Pengujian ini menggunakan *tool* Burp Suite untuk mencari kombinasi *username* dan *password* yang benar untuk masuk ke sistem. Setelah berhasil menangkap status dan proses *login* ini pada sistem bisa dilakukan penetrasi *password*. Hasil pengujian dapat dilihat pada Gambar 8. Pengujian percobaan serangan *broken authentication* pada *tool* Burp Suite menghasilkan data yang dikirimkan berupa blok *hash* atau terenkripsi, sehingga data yang ada dapat terjamin keamanannya dan terjaga rahasianya.



Gambar 8. Capture data setelah menggunakan *Blockchain*.

Gambar 8 menunjukkan hasil *capture* data setelah menggunakan *blockchain*. Data yang didapatkan berupa blok yang terenkripsi. *Username* dan *password* yang dimasukkan sebelumnya berhasil diubah. Serangan dengan menggunakan Burp Suite tidak dapat mendeteksi data dari pengguna setelah menggunakan teknologi *blockchain.*

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, kesimpulan yang dapat diambil yaitu sistem *login* berhasil dibangun. Dengan menggunakan teknologi *blockchain*, data dari *user* terjaga kerahasiaannya. Pengujian dilakukan dengan menggunakan Burp Suite. Data yang dikirimkan berupa blok *hash* dan terenkripsi, sehingga data terjamin keamanannya. Pengujian keamanan sistem *login* ini berhasil dilakukan dengan tepat sehingga sistem lebih aman daripada sebelumnya. Burp Suite sebagai pengujian sistem *login* lebih spesifik dalam melakukan pengujian keamanan. Data *username* dan *password* diubah menjadi chiperteks sehingga penyerang tidak dapat mengetahui data dari pengguna.

DAFTAR PUSTAKA

Azis, H., & Fattah, F. (2019). ANALISIS LAYANAN KEAMANAN SISTEM KARTU TRANSAKSI ELEKTRONIK MENGGUNAKAN METODE PENETRATION TESTING. *ILKOM Jurnal Ilmiah*, *11*(2), 167–174. https://doi.org/10.33096/ilkom.v11i2.447.167-174

Bouscaren, E. (1989). Elementary pairs of models. *Annals of Pure and Applied Logic*, *45*(2), 129–137. https://doi.org/10.1016/0168-0072(89)90057-2

Dilley, J., Poelstra, A., Wilkins, J., Piekarska, M., Gorlick, B., & Friedenbach, M. (2016). *Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks*.

Fadlil, A., Riadi, I., & Nugrahantoro, A. (2020). Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *11*(3), 155. https://doi.org/10.24843/LKJITI.2020.v11.i03.p04

Fauzan, N. I. (2018). TEKNOLOGI BLOCKCHAINDAN PERANANNYA DALAM ERA DIGITAL. *Jurnal BJB University*, *4*, 1–15.

Guntoro, G., Costaner, L., & Musfawati, M. (2020). ANALISIS KEAMANAN WEB SERVER OPEN JOURNAL SYSTEM (OJS) MENGGUNAKAN METODE ISSAF DAN OWASP (STUDI KASUS OJS UNIVERSITAS LANCANG KUNING). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *5*(1), 45. https://doi.org/10.29100/jipi.v5i1.1565

Harjowinoto, D., Noertjahyana, A., & Andjarwirawan, J. (2016). VULNERABILITY TESTING PADA SISTEM ADMINISTRASI RUMAH SAKIT X. *Jurnal Infra*, *4*(1), 227–232.

Hassan, M. M., Nipa, S. S., Akter, M., Haque, R., Deepa, F. N., Rahman, M. M., Siddiqui, M., & Sharif, M. H. (2018). Broken Authentication and Session Management Vulnerability: A Case Study of Web Application. *International Journal of Simulation: Systems, Science & Technology*, 1–11. https://doi.org/10.5013/IJSSST.a.19.02.06

Hu, S. D. K., Palit, H. N., & Handojo, A. (2019). IMPLEMENTASI BLOCKCHAIN: STUDI KASUS E-VOTING. *Jurnal Infra*, *7*(1), 183–189.

Joshi, C., & Kumar, U. (2016). Security Testing and Assessment of Vulnerability Scanners in Quest of Current Information Security Landscape. *International Journal of Computer Applications*, *145*(2), 1–7. https://doi.org/10.5120/ijca2016910563

Kunang, Y. N., Muklis, F., & Sauda, S. (2013). PENGUJIAN CELAH KEAMANAN PADA CMS (CONTENT MANAGEMENT SYSTEM). *Prosiding Seminar Nasional Ilmu Komputer (SeNAIK 2013)*, 398–406.

Laksmiati, D. (2020). VULNERABILITY ASSESSMENT PADA SITUS WWW.HATSEHAT.COM MENGGUNAKAN OPENVAS. *Jurnal Akrab Juara*, *5*(3), 240–246.

OWASP. (2017). *OWASP Top Ten Web Application Security Risks*. OWASP. https://owasp.org/www-project-top-ten/

Pangalila, R., Noertjahyana, A., & Andjarwirawan, J. (2015). PENETRATION TESTING SERVER SISTEM INFORMASI MANAJEMEN DAN WEBSITE UNIVERSITAS KRISTEN PETRA. *Jurnal Infra*, *3*(2), 271–276.

Parizi, R. M., Dehghantanha, A., Choo, K.-K. R., & Singh, A. (2018). Empirical Vulnerability Analysis of Automated Smart Contracts Security Testing on Blockchains. *In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering (CASCON18)*, 103–113. https://doi.org/10.5555/3291291.3291303

Putra, A. W. P., Bhawiyuga, A., & Data, M. (2018). Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, *2*(2), 584–593.

Putra, H. F., Wirawan, W., & Penangsang, O. (2019). Penerapan Blockchain dan Kriptografi untuk

Keamanan Data pada Jaringan Smart Grid. *Jurnal Teknik ITS*, *8*(1). https://doi.org/10.12962/j23373539.v8i1.38525

Rahardja, U., Harahap, E. P., & Christianto, D. D. (2019). PENGARUH TEKNOLOGI BLOCKCHAIN TERHADAP TINGKAT KEASLIAN IJAZAH. *Technomedia Journal*, *4*(2), 211–222. https://doi.org/10.33050/tmj.v4i2.1107

Ramadhan, M. S., & Ariyani, F. (2018). PENINGKATAN KEAMANAN LOGIN WEBSITE DENGAN IMPLEMENTASI ONE TIME PASSWORD MENGGUNAKAN ALGORITMA SHA1 DAN MD5 BERBASIS MOBILE. *SKANIKA*, *1*(2), 689–696.

Riadi, I., Umar, R., & Busthomi, I. (2020). Optimasi Keamanan Autentikasi dari Man in the Middle Attack (MiTM) Menggunakan Teknologi Blockchain. *Journal of Information Engineering and Educational Technology*, *4*(1), 15–19. https://doi.org/http://dx.doi.org/10.26740/jieet.v4n1.p15-19

Riadi, I., Umar, R., & Lestari, T. (2020). Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *5*(3), 146–152. https://doi.org/10.14421/jiska.2020.53-02

Riadi, I., Yudhana, A., & W, Y. (2020). Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *7*(4), 853. https://doi.org/10.25126/jtiik.2020701928

Rusdan, M., & Sabar, M. (2020). Design and Analysis of Wireless Network with Wireless Distribution System using Multi-Factor Authentication-based User Authentication. *Journal of Information Technology*, *2*(1), 17–24. https://doi.org/10.47292/joint.v2i1.004

Sai Kiran, K. V. V. N. L., Devisetty, R. N. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques. *Procedia Computer Science*, *171*(2019), 2372–2379. https://doi.org/10.1016/j.procs.2020.04.257

Sitinjak, H. S. F., Hediyanto, U. Y. K. S., & Widjajarto, A. (2020). Security Auditing Pada Vulnerable Machine Menggunakan Open Source Ids Dan Vulnerability Scanner Berdasarkan Nist Cybersecurity Framework. *EProceedings of Engineering*, *7*(2), 7638–7646.

Sudiarto Raharjo, W., E.K. Ratri, I. D., & Susilo, H. (2017). IMPLEMENTASI TWO FACTOR AUTHENTICATION DAN PROTOKOL ZERO KNOWLEDGE PROOF PADA SISTEM LOGIN. *Jurnal Teknik Informatika Dan Sistem Informasi*, *3*(1), 127–136. https://doi.org/10.28932/jutisi.v3i1.579

T, G. S., & Sasikala, D. (2019). Vulnerability Assessment of Web Applications using Penetration Testing. *International Journal of Recent Technology and Engineering*, *8*(4), 1552–1556. https://doi.org/10.35940/ijrte.B2133.118419

Wibowo, F., Harjono, H., & Wicaksono, A. P. (2019). Uji Vulnerability pada Website Jurnal Ilmiah Universitas Muhammadiyah Purwokerto Menggunakan OpenVAS dan Acunetix WVS. *Jurnal Informatika*, *6*(2), 212–217. https://doi.org/10.31311/ji.v6i2.5925