

Implementasi *Private Blockchain* Menggunakan Multichain Sebagai Sistem Keamanan Data IoT

Lalu Ocky Saktiya Luhung¹, Misbahuddin¹ Giri Wahyu Wiriasto¹

¹Jurusan Teknik Elektro – Universitas Mataram, Jalan Majapahit 62, Kota Mataram - 82115, Indonesia

ARTICLE INFO

Article history (8 pt):

Received

Revised

Accepted

Keywords (8 pt):

Blockchain

Internet of Things(IoT)

Private Blockchain;

Hashing

Multichain

ABSTRACT

Integrasi teknologi blockchain dengan Internet of Things (IoT) untuk meningkatkan keamanan dan privasi data menjadi solusi yang semakin relevan di era digital. Dengan fokus pada keamanan IoT saat mendistribusikan data, risiko manipulasi data dapat dikurangi karena blockchain memungkinkan struktur terdesentralisasi dan terenkripsi. Integritas data ditingkatkan melalui verifikasi transaksi dan penyimpanan data yang diotorisasi dengan hash value dan bukti transaksi (txid). Penggunaan jaringan private blockchain mengontrol akses data, meningkatkan keamanan, privasi, dan memastikan akses terbatas pada pihak yang diizinkan. Percobaan dilakukan dan didapatkan nilai kemungkinan kumulatif berdasarkan jumlah block dengan nilai kemungkinan kumulatif dihasilkan dari nilai block yang bertambah setiap 15 detik sampai mendapatkan jumlah 6 konfirmasi dari prinsip blockchain sehingga suatu block dinyatakan valid dalam jaringan blockchain, yang mengartikan setelah block ditambahkan ke rantai dibutuhkan 6 block tambahan untuk memastikan keamanan transaksi. Kemudian didapatkan juga nilai kemungkinan kumulatif berdasarkan jumlah waktu yang dibutuhkan membuat block di dalam blockchain, dengan nilai kemungkinan kumulatif dihasilkan dari nilai block yang bertambah setiap 15 detik sampai mendapatkan jumlah 6 konfirmasi dari prinsip blockchain sehingga suatu block dinyatakan valid dalam jaringan blockchain. Kemudian hingga suatu block dinyatakan valid saat waktu mencapai 60 detik sampai 165 detik. Hasilnya menunjukkan bahwa blockchain dapat meningkatkan keamanan data IoT serta menjaga integritas data dengan lebih baik, memperkuat otorisasi, dan mengontrol akses data secara lebih selektif

corresponding Author:

Lalu Ocky Saktiya, Jurusan Teknik Elektro – Universitas Mataram, Jalan Majapahit 62, Kota Mataram – 83115, Indonesia

Email: laluocky450@gmail.com

1. INTRODUCTION

Dengan kemajuan teknologi komunikasi dan pengenalan jaringan 5G di mana-mana, teknologi *Internet of Things* mulai berkembang pada tingkat yang eksponensial. *Smart Home*, *Smart City*, *e-Health*, dan *Internet of Things* untuk perusahaan industri, intelijen terdistribusi, dan sistem lainnya adalah cara yang efektif dan akrab bagi masyarakat untuk meningkatkan banyak proses, misalnya, proses untuk monitoring keadaan rumah berdasarkan sensor dan proses lain yang dapat menjadi otomatis. Pendekatan proses seperti itu mengurangi pengaruh faktor manusia dan berkontribusi pada peningkatan efisiensi perusahaan, di mana ada semua prasyarat untuk penggunaan teknologi IoT. Terlepas dari semua efektivitas dan prevalensinya, teknologi *Internet of Things* memiliki banyak tantangan dan masalah yang terkait dengan keamanan dan konfigurasi perangkat IoT yang aman. Keberadaan sejumlah besar perangkat semacam itu penuh dengan bahaya, karena penyerang dapat mengendalikannya dan mengatur serangan DDoS dan manipulasi lalu lintas lainnya menggunakan perangkat IoT, yang mengirim perangkat ini ke server. Salah satu contoh serangan terpadu pada beberapa perangkat IoT adalah botnet. Botnet adalah kumpulan perangkat yang disusupi di

bawah kendali penyerang. Mirai adalah worm dan botnet yang dibentuk oleh perangkat yang diretas (disusupi) seperti *Internet of Things* (pemutar video, webcam pintar, dll.). Botnet ini meretas perangkat dengan menebak kata sandi untuk port 23 (telnet). Dalam sistem IoT terpusat, terkadang cukup untuk meretas server atau mikrokontroler yang bertanggung jawab untuk komunikasi antara sekelompok besar perangkat agar dapat mengontrol semua perangkat yang berkomunikasi melalui protokol terpusat dengan server yang dikompromikan [14].

Di lain sisi, perkembangan teknologi yang cukup maju memungkinkan untuk mengurangi dampak dari masalah yang ada, bahkan ada kemungkinan dapat menyelesaikan masalah yang ada. Teknologi yang dimaksud adalah *blockchain*. Pendekatan penyimpanan yang terdesentralisasi untuk menyediakan penyimpanan data terdapat pada teknologi *blockchain* dan layanan berbagi. Untungnya, sifat teknologi *blockchain* dapat memberikan solusi yang baik untuk sistem penyimpanan yang terdesentralisasi. *Blockchain* terdiri dari blok-blok individual yang dihubungkan oleh fungsi *hash*, dan setiap blok berisi nilai *hash* dari blok sebelumnya, time-stamp, dan data transaksi. *Blockchain* dapat dianggap sebagai *database* buku besar terdistribusi, yang terdesentralisasi, terbuka dan transparan, anti-rusak, dan dapat dilacak, serta menyediakan metode penyimpanan yang aman dan andal untuk data.

Pada dasarnya *Blockchain* adalah buku besar basis data yang terdesentralisasi, terdistribusi saling berbagi dan sangat sulit untuk diubah yang menyimpan daftar aset dan transaksi di jaringan *peer-to-peer*, serta telah merantai blok data yang telah diberi cap waktu dan divalidasi oleh *miners* [13]. Algoritma *Proof of work* adalah algoritma yang paling banyak digunakan. Algoritma ini digunakan oleh mata uang kripto seperti bitcoin dan ethereum, masing-masing memiliki perbedaan tersendiri. Algoritma PoW digunakan untuk mengkonfirmasi transaksi dan menghasilkan *block* baru di *blockchain* [10]. *Blockchain* menampilkan *database* yang terdesentralisasi dan tidak dapat rusak yang memiliki potensi tinggi untuk beragam penggunaan. *Blockchain* adalah *database* terdistribusi yang banyak digunakan untuk mencatat transaksi yang berbeda, setiap data yang dimasukkan akan di-*encrypt* sehingga tidak memudahkan orang lain untuk memanipulasi dan menggandakan sertifikat tersebut. Setelah konsensus tercapai di antara *node* yang berbeda, transaksi ditambahkan ke *block* yang sudah menyimpan catatan beberapa transaksi. Setiap *block* berisi nilai *hash* dari pasangan terakhirnya untuk koneksi sehingga *node* menjaga *database* bersama-sama. Di bawah *blockchain*, sebuah *block* menjadi tervalidasi hanya setelah diverifikasi oleh banyak pihak. Selain itu, data dalam *block* tidak dapat dimodifikasi secara sewenang-wenang [8]. Kriptografi merupakan seni dan ilmu untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan untuk orang yang hanya memiliki sebuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan [2]. Teknologi *blockchain* telah diramalkan oleh industri dan komunitas penelitian sebagai teknologi yang sangat menyita perhatian yang siap memainkan peran utama mengelola, mengendalikan, dan yang paling penting mengamankan perangkat IoT. *Blockchain* menggunakan algoritma *hash* ing SHA-256 untuk memberikan bukti kriptografi yang kuat untuk otentikasi dan integritas data. *Blockchain* memiliki riwayat penuh dari semua transaksi dan memberikan kepercayaan terdistribusi global. Salah satu tujuan penggunaan *blockchain* adalah untuk menghilangkan pihak ketiga atau *Trusted Third Parties* (TTP). TTP atau otoritas dan layanan terpusat dapat diganggu, ditembus kemanannya, dan diretas. Mereka juga dapat berbuat jahat dan berprilaku korup di masa depan, meskipun mereka dapat dipercaya sekarang [13].

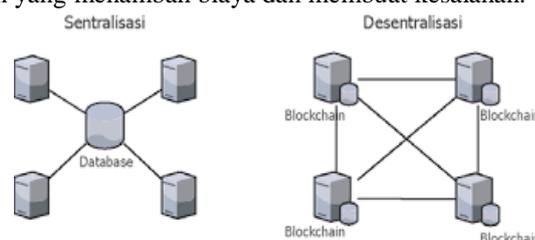
1. Internet of Things (IoT)

Internet of Things (IoT), merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. Misalnya sebuah *Smart Home* yang dapat dikelola lewat *smartphone* dengan bantuan koneksi internet. Pada dasarnya IoT bila mendapatkan sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima untuk dianalisa [6].

Dalam IoT, asosiasi dari berbagai perangkat heterogen mengurangi kemampuan jaringan sumber daya, yang menarik perhatian para peneliti ke arah bidang yang sedang berkembang ini. Beberapa aplikasi membutuhkan transmisi yang aman, sementara beberapa aplikasi lainnya membutuhkan penyimpanan lokal untuk transmisi cepat dan waktu respons yang rendah. Sejumlah besar konten ini dengan pemrosesan lokal akan membutuhkan teknik yang canggih untuk administrasi lokal [20].

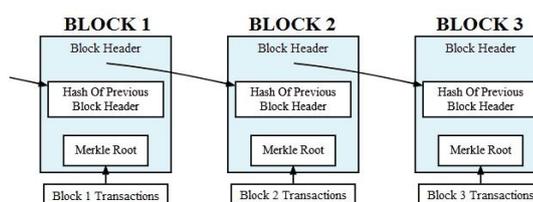
2. Blockchain

Blockchain adalah sebuah database terdistribusi atau buku besar yang dibagikan di antara *node-node* jaringan komputer. *Blockchain* terkenal dengan peran pentingnya dalam sistem mata uang digital untuk menjaga catatan transaksi yang aman dan terdesentralisasi, tetapi tidak terbatas pada penggunaan mata uang digital. *Blockchain* dapat digunakan untuk membuat data dalam industri apa pun menjadi tidak dapat diubah, istilah yang digunakan untuk menggambarkan ketidakmampuan untuk diubah. Karena tidak ada cara untuk mengubah sebuah blok, satu-satunya kepercayaan yang dibutuhkan adalah pada saat pengguna atau program memasukkan data. Aspek ini mengurangi kebutuhan akan pihak ketiga yang terpercaya, yang biasanya adalah auditor atau manusia lain yang menambah biaya dan membuat kesalahan.



Gambar 1. Blockchain dan Database

Gambar 1 merupakan gambaran perbedaan antara penyimpanan yang bersifat sentralisasi yang digunakan pada *database* dan penyimpanan yang bersifat desentralisasi yang digunakan pada *blockchain*. *Blockchain* didistribusikan, yang berarti banyak salinan disimpan di banyak mesin, dan semuanya harus cocok agar *valid*. *Blockchain* mengumpulkan informasi transaksi dan memasukkannya ke dalam sebuah blok, seperti sebuah sel pada spreadsheet yang berisi informasi. Setelah penuh, informasi tersebut dijalankan melalui algoritma enkripsi, yang menciptakan angka heksadesimal yang disebut *hash*. *Hash* tersebut kemudian dimasukkan ke dalam header blok berikutnya dan dienkripsi dengan informasi lain dalam blok tersebut. Hal ini menciptakan serangkaian blok yang dirantai bersama [9]. *Blockchain* adalah sebuah inovasi yang telah menarik banyak sekali pertimbangan dari para ahli dan *Blockchain* adalah sebuah blok catatan yang diikat. Setiap blok berisi dua bagian penting yaitu pertukaran dan header blok. Dalam *blockchain*, sebuah pertukaran membentuk korespondensi penting yang mengizinkan dua hub untuk memperdagangkan data satu sama lain. Struktur esensial dari pertukaran muncul di sisi kiri. Perhatikan bahwa berbagai peluncuran *blockchain* mungkin memiliki sedikit variasi dalam struktur pertukaran seperti pada Gambar 2 dibawah ini [17].



Gambar 2. Struktur Blockchain

Blockchain

Public Blockchain (Blockchain Publik) memungkinkan setiap orang untuk bisa bergabung. Sebagaimana namanya, publik, *Blockchain* ini ditujukan untuk umum/publik. *Blockchain* ini terbuka untuk semua orang, di mana setiap orang bisa menjadi simpul (*node*), bisa membaca, menulis dan melakukan *update* pada *blockchain* dengan membuat alamat pribadi (*personal address*)-nya sendiri. Dengan menggunakan kunci pribadi (*private key*) yang telah diubah menjadi kunci publik (*public key*) memungkinkan siapa saja yang memiliki koneksi internet dan perangkat komputasi yang bisa menjalankan perangkat lunak *Blockchain* untuk dapat berpartisipasi [21]. *Private Blockchain (Blockchain Pribadi)*, pemilik *Blockchain* memiliki pengaruh yang signifikan terhadap desain dan operasi selanjutnya [21]. *Private Blockchain* adalah jenis *Blockchain* yang bersifat tertutup dan bertujuan untuk melakukan pertukaran informasi secara internal saja, sehingga pihak-pihak yang tidak bergabung tidak dapat melihat proses-proses apa saja yang dilakukan pada *Blockchain* tersebut [4]. Dalam tipe ini, jika ada yang ingin menjadi *node*, mereka harus mendapatkan ijin dari otoritas (pemilik) *Blockchain* [21].

Algoritma Consensus

Algoritma *consensus* adalah sebuah teknik untuk mencapai sebuah kesepakatan bersama di dalam sebuah kelompok. Konsensus *Blockchain* adalah strategi untuk membuat keseragaman dan kesopanan di dunia online. Kerangka kerja konsensus yang digunakan untuk ini pemahaman ini dikenal sebagai hipotesis *consensus* [15]. *Blockchain* menerapkan beberapa algoritma *consensus* seperti berikut :

Proof of Work

Proof of Work (PoW) adalah strategi konsensus yang digunakan pada Bitcoin. Jika sebuah *node* ingin mencatat sebuah blok, banyak usaha yang harus dilakukan oleh *node* tersebut untuk membuktikan bahwa *node* tersebut tidak memiliki keinginan untuk menyerang jaringan *blockchain* yang ada, hal ini yang mendasari cara kerja konsensus ini. Konsensus ini membutuhkan nilai *hash* yang dihitung tersebut untuk sama dengan atau lebih kecil dari nilai yang telah ditentukan sebelumnya. Ketika salah satu *node* dalam jaringan berhasil mencapai nilai yang ditentukan, maka blok tersebut akan disebar ke jaringan dan semua *node* dalam jaringan masing-masing mengkonfirmasi kebenaran nilai *hash* itu, dan setelah itu blok dinyatakan valid. Setelah itu semua *node* harus menambahkan blok ini ke *blockchain* mereka. *Nodes* yang menghitung nilai *hash* ini disebut dengan *miners* dan proses pengerjaan PoW ini disebut *mining* dalam Bitcoin [11].

Proof of Stake

Proof of Stake (PoS) adalah protokol yang lebih ramah energi dibandingkan dengan *Proof of Work* (PoW). *Miner* dalam PoS harus membuktikan kepemilikan dengan memiliki sejumlah uang (*cryptocurrency* yang dibuat pada *blockchain* tertentu). Pemilihan dengan melihat jumlah saldo cukup tidak adil karena orang yang paling kaya di jaringan tersebut akan mendominasi. Oleh karena itu, terdapat beberapa solusi yang diajukan untuk mengombinasikan jumlah saldo dan hal lain untuk menambah blok baru pada jaringan. Contoh, pada *blockchain*, dimana membuat blok selanjutnya akan diacak dengan menggunakan rumus yang mencari nilai *hash* uang paling kecil lalu dikombinasikan dengan saldo orang tersebut [12].

Raspberry Pi

Raspberry Pi adalah sebuah komputer kecil yang mumpuni yang memiliki ukuran sebesar kartu ATM. Perangkat ini dapat digunakan untuk proyek elektronik karena memiliki input, output, port digital dan dapat melakukan banyak hal layaknya PC desktop atau komputer. *Raspberry Pi* dapat menghubungkannya ke TV atau layar komputer dan keyboard. *Raspberry Pi* dibuat di Inggris oleh *Raspberry Pi Foundation* [7]. *Raspberry Pi* tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka-panjang. *Raspberry Pi* memiliki beberapa tipe, yaitu Model A dan Model B, dengan perbedaan spesifikasi seperti harga, SoC, CPU, GPU, RAM, dan lain-lain.[22] Pada penelitian ini, menggunakan *Raspberry Pi 3 Model B* [7].

Multichain

Multichain adalah platform yang membantu pengguna untuk membangun *Blockchain* pribadi tertentu yang dapat digunakan oleh organisasi untuk transaksi. API sederhana yang disediakan Multichain membantu untuk mengatur rantai. Tujuan Multichain membuat visibilitas *blockchain* harus selalu secara aktif disimpan dalam peserta yang dipilih untuk menghindari kebingungan untuk memastikan stabilitas dan kontrol atas transaksi, dan proses penambangan (*mining*) dapat dilakukan dengan lebih aman. Model *blockchain* ini hanya mentransaksikan akun yang divalidasi ke peserta rantai ini. Dalam Multichain terdapat Proses *hand-shaking* dimana terjadi ketika *node* dalam *blockchain* terhubung satu sama lain. Multichain terjadi ketika dua *node blockchain* terhubung. Identitas setiap *node* mewakili dirinya sendiri dengan alamat dengan daftar izin. Oleh karena itu, setiap *node* yang diwakilinya mengirimkan pesan ke pengguna lain. Koneksi *Peer to Peer* (P2P) dibatalkan jika mereka tidak menerima hasil yang memuaskan dari proses tersebut [18].

Related Work

Pada tinjauan pustaka ini akan membahas penelitian yang pernah dilakukan sebelumnya atau yang relevan dengan penelitian yang akan dilakukan sebagai acuan dalam pengerjaan penelitian. Penelitian pertama *Blockchain for IoT Security and Privacy: The Case Study of a Smart Home*. Studi ini mempelajari penggunaan *blockchain* untuk meningkatkan keamanan dan privasi pada lingkungan *Smart Home*, yang merupakan salah satu aplikasi penting dari IoT. Pendekatan studi kasus untuk mengevaluasi keefektifan

penggunaan *blockchain* dalam meningkatkan keamanan dan privasi pada *Smart Home*. Penggunaan *blockchain* pada *Smart Home* dapat memberikan beberapa keuntungan dalam hal keamanan dan privasi, seperti mengurangi risiko serangan peretas dan memastikan privasi data pengguna. Secara keseluruhan, penelitian ini menyimpulkan bahwa penggunaan *blockchain* dapat memberikan solusi yang efektif untuk mengatasi tantangan keamanan dan privasi pada *Smart Home*. Studi ini menunjukkan bahwa implementasi *blockchain* pada *Smart Home* dapat meningkatkan tingkat keamanan, privasi, dan pengelolaan data yang lebih efektif pada lingkungan IoT [5].

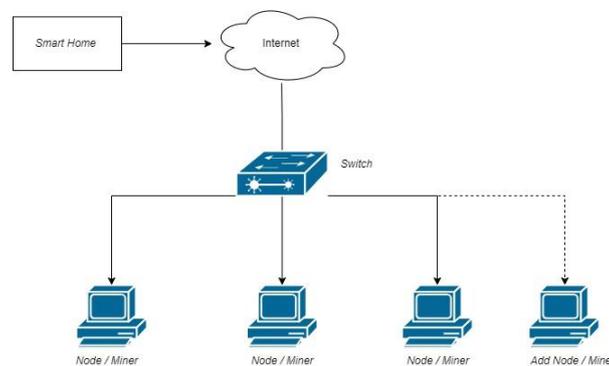
Penelitian kedua, *Blockchain Technology Implementation in Raspberry Pi for Private Network*. Penelitian ini mencoba membuktikan perangkat komputer yang murah seperti komputer mini dapat digunakan dalam pengembangan teknologi *blockchain*. Teknologi komputer mini digunakan karena banyaknya permasalahan yang terjadi akibat penggunaan sumber daya perangkat keras yang dibahas besar, penelitian ini menggunakan desain eksperimental. Eksperimen dilakukan dengan perangkat yang digunakan dan teknologi *blockchain*. Perangkat yang digunakan adalah komputer mini *Raspberry Pi*, dan teknologi *blockchain* adalah platform Ethereum. Hasil dari penelitian ini menunjukkan proses instalasi hingga sistem *blockchain* dapat berjalan. Hasil penelitian ditunjukkan dengan grafik evaluasi performa yang menunjukkan parameter *commulative probability* akan meningkat apabila *block number* bertambah [7].

Penelitian ketiga, *Security threats and solutions to IoT using Blockchain: A Review*. Penelitian ini mengevaluasi penggunaan teknologi *blockchain* sebagai solusi untuk mengatasi ancaman keamanan pada lingkungan *Internet of Things* (IoT). Penelitian ini mengidentifikasi beberapa ancaman keamanan yang dihadapi oleh lingkungan IoT, seperti serangan DDoS, pencurian data, dan manipulasi data. Kemudian, penelitian ini membahas bagaimana teknologi *blockchain* dapat membantu mengatasi masalah-masalah keamanan ini. Dengan penggunaan *blockchain*, sistem IoT dapat dilindungi dari serangan peretas dan manipulasi data. Selain itu, penelitian ini juga membahas tentang tantangan dan kelemahan penggunaan *blockchain* untuk sistem IoT, seperti keterbatasan kapasitas transaksi dan biaya operasional yang tinggi. Namun, penelitian ini menyimpulkan bahwa penggunaan *blockchain* dapat memberikan solusi yang efektif untuk mengatasi ancaman keamanan pada lingkungan IoT. Secara keseluruhan, penelitian ini memberikan informasi yang berguna tentang penggunaan teknologi *blockchain* untuk mengamankan sistem IoT. Studi ini memberikan wawasan tentang bagaimana *blockchain* dapat digunakan untuk mengatasi ancaman keamanan dan memberikan solusi yang efektif untuk mengamankan data pada lingkungan IoT [1].

2. METHOD

Pada penelitian ini menggunakan topologi jaringan, yaitu terdapat jaringan internet untuk mejadi jaringan lokal antara perangkat *Smart Home* dengan PC (*node / miner*) serta terdapat 3 *node* yang dapat berperan sebagai *miner* dan saling terhubung antara *node* yang satu dengan *node* yang lain. Perangkat *Smart Home* yang akan disimulasikan dalam bentuk program Python yang dijalankan pada perangkat *Raspberry Pi* yang akan mengirimkan data kepada PC yang berperan sebagai *miner*.

Topologi Jaringan

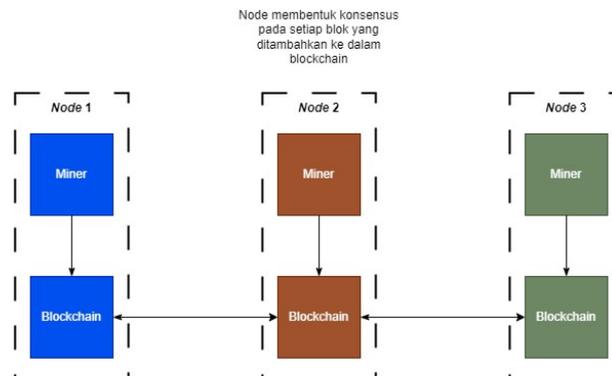


Gambar 3. Topologi Jaringan

Gambar 3 merupakan topologi jaringan yang membentuk topologi star dimana pada topologi tersebut PC berperan sebagai *node* dalam jaringan dan dapat juga berperan sebagai *miner*. *Miner* bertanggung jawab

untuk melakukan pengecekan dan validasi terhadap aktivitas yang terjadi di dalam sistem, misalnya *Smart Home* devices yang hendak mengirimkan dan menyimpan data ke dalam sistem apakah diizinkan atau tidak dan apakah data yang dikirimkan merupakan data yang valid atau tidak. Penelitian ini menggunakan platform Multichain yang akan dipasang pada semua *node* yang terdapat di dalam sistem.

Alur Komunikasi Blockchain

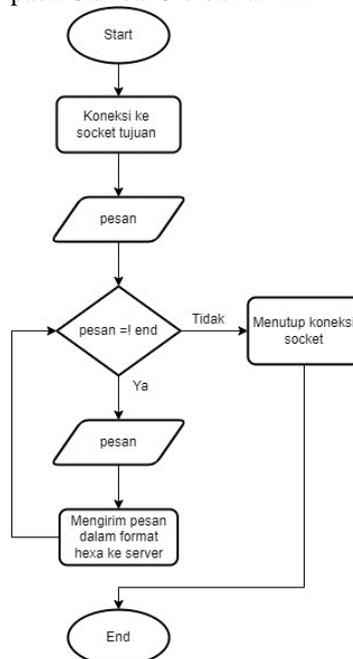


Gambar 4. Alur Komunikasi Blockchain

Gambar 4 merupakan alur komunikasi *blockchain* yang terjadi pada 3 buah *node*. Peran *node* dan *miner* dalam jaringan *blockchain* yaitu *node* dapat saling berkomunikasi dengan sesama *node* karena sudah terjadi koneksi secara *peer-to-peer* antar *node*, dimana masing-masing *node* terdapat *miner* untuk memvalidasi transaksi dan menambahkan ke blok baru ke dalam *blockchain*.

Perangkat Smart Home

Pada penelitian ini dilakukan pembuatan perangkat *Smart Home* berbasis program yang dimana pembuatan program menggunakan bahasa python. *Smart Home* berbasis program ini akan dijalankan pada perangkat *Raspberry Pi* dengan tujuan untuk dapat mengirimkan data ke *node* admin yang kemudian akan diproses dan akan didistribusikan ke *blockchain* apabila data sudah di cek dan divalidasi kemudian digambarkan dengan diagram alir seperti pada Gambar 5 dibawah ini.



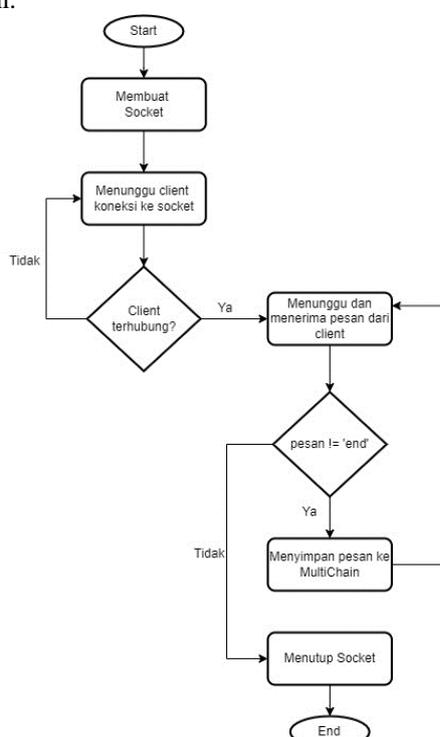
Gambar 5. Diagram Alir Perangkat Smart Home

Pengiriman Data

Data atau pesan dikirimkan dari perangkat *Smart Home* yang telah dihubungkan dengan *node* admin kemudian perangkat *Smart Home* akan disimulasikan dengan menggunakan program Python pada perangkat *Raspberry Pi* yang mengirimkan data kepada program server pada *node* admin. Setelah terhubung, perangkat *Smart Home* dapat mengirim data ke *node* Multichain menggunakan *socket*. Data akan diserialisasi menjadi format yang sesuai *Hexadesimal* atau format khusus, lalu kirim data melalui koneksi *socket* yang sudah dibuat. Data yang dikirimkan akan memiliki waktu jeda tiap pengiriman data adalah 25 detik karena harus menunggu proses *mining* dari *blockchain* terlebih dahulu.

Penerimaan Data

Data yang telah dikirim oleh perangkat *Smart Home* kemudian akan diterima oleh sebuah program yang dijalankan pada *node* admin. Program ini memiliki fungsi untuk menerima data yang dikirim oleh perangkat *Smart Home* kemudian meneruskannya kepada Multichain kemudian digambarkan dengan diagram alir seperti pada Gambar 6 dibawah ini.



Gambar 6. Alur Penerimaan Data

3. Penyimpanan Data

Setelah terhubung dengan API milik Multichain, maka untuk selanjutnya data dapat diterima dan disimpan ke dalam *blockchain*. Data yang telah diterima dan masih dalam format *hexadesimal* akan diteruskan ke dalam *blockchain* untuk disimpan ke dalam blok baru. Multichain memiliki fitur *stream* yang berfungsi sebagai tempat untuk menyimpan data secara umum, dan istilah menyimpan data di dalam stream dikenal dengan istilah *publish*.

4. Proses Validasi

Ada dua jenis media yang akan mencoba mengirimkan data ke dalam *blockchain* yaitu *permission ed device* dan *permission less device*. *Permission ed devices* adalah perangkat yang diizinkan oleh sistem untuk menyimpan dan melihat data. Sedangkan *permission less devices* adalah perangkat yang berada di dalam maupun di luar sistem yang tidak memiliki izin apapun untuk melakukan aktivitas yang sama seperti *permission ed devices*.

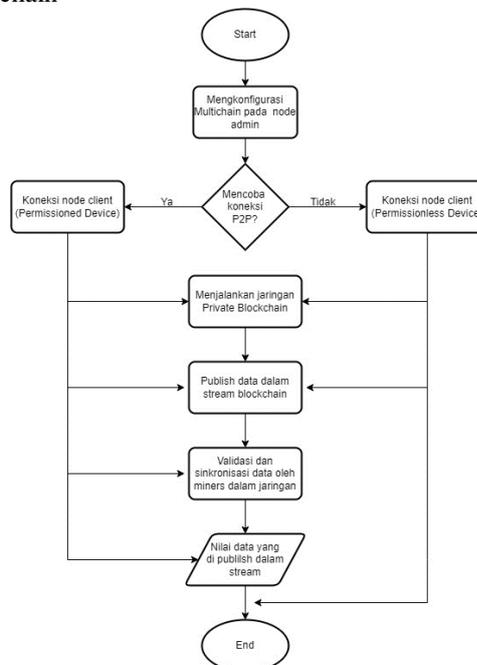
Proses validasi data dalam Bitcoin biasa disebut dengan proses *mining*. Bitcoin menggunakan konsep yang bernama PoW (*Proof of Work*). Proses validasi didalam multichain menggunakan skema kombinasi

antara jumlah *miner* dengan parameter bernama *mining-diversity* yang dibatasi dengan $0 \leq \text{mining diversity} \leq 1$. *Miner* adalah sebutan untuk komputer atau *node* yang bertugas untuk melakukan proses validasi data pada *blockchain*.

1. Menerapkan semua perubahan izin yang ditentukan oleh transaksi dalam blok secara berurutan.
2. Menghitung jumlah *miner* yang diizinkan yang ditentukan setelah menerapkan perubahan, misalkan jumlah *miner* adalah 2.
3. Mencari nilai *spacing* dengan mengkalikan jumlah *miner* dengan nilai *mining-diversity*. Misalkan nilai *mining-diversity* = 0.3, maka diperoleh nilai *spacing* = $2 \times 0.3 = 0.6$. Kemudian nilai 0.6 dibulatkan ke atas menjadi 1 sehingga nilai *spacing* yang sebenarnya adalah 1.
4. Jika *miner* dari *block* ini telah melakukan validasi pada salah satu dari (nilai *spacing* -1) *block* sebelum ini, maka proses validasi tidak sah. Jika tidak, maka *miner* yang bersangkutan akan didelegasikan untuk melakukan proses validasi pada *block* tersebut dan *block* akan dianggap valid dan dapat disimpan ke dalam *blockchain*.

Skema ini memberlakukan teknik penjadwalan *round-robin*, dimana *miner* yang diizinkan harus membuat *block* secara bergiliran untuk menghasilkan *blockchain* yang valid. Parameter *mining-diversity* mendefinisikan ketatnya skema Nilai 1 pada *mining-diversity* memastikan bahwa setiap *miner* akan masuk ke dalam rotasi *round-robin*, sedangkan nilai 0 menunjukkan tidak ada batasan sama sekali. Secara umum nilai yang lebih tinggi akan lebih aman, namun nilai yang terlalu dekat dengan 1 dapat menyebabkan *blockchain* membeku jika beberapa *miner* menjadi tidak aktif.

- Diagram alir Pengujian Multichain



Gambar 7. Diagram Alir Pengujian

3. RESULTS AND DISCUSSION

Langkah pertama pada implementasi ini adalah melakukan konfigurasi dan instalasi Multichain pada setiap *node*. Kemudian diatur 1 *node* sebagai *server* atau *node admin* dan *node* lainnya disebut sebagai *node client* yang kemudian *node client* akan dibedakan perannya yaitu sebagai *permissioned device* dan *permissionless device*. Multichain diinstal pada komputer yang berjalan dengan *operation system* (OS) menggunakan Ubuntu 22.04.

5. Skenario Permissioned Device

Koneksi antar *node* didalam jaringan *private blockchain* dapat terjadi dengan proses koneksi menggunakan mekanisme *hand shaking*, dimana *node* yang ingin terhubung (*Client*) akan diberi alamat

secara khusus, alamat tersebut digunakan untuk terhubung dengan *node* yang kemudian akan memberikan *permission* (admin).

1. Koneksi oleh *Node Client*

```
Retrieving blockchain parameters from the seed node 192.168.9.254:4287
...
Blockchain successfully initialized.

Please ask blockchain admin or user having activate permission to let
you connect and/or transact:
multichain-cli iotchain grant 1Ef2MZGZBj7vC36g7bGtmXopDsAnLfpCjuHGeP
connect
multichain-cli iotchain grant 1Ef2MZGZBj7vC36g7bGtmXopDsAnLfpCjuHGeP
connect, send, receive
```

Gambar 8. Koneksi *Node Client*

Gambar 8 merupakan *node client* yang mencoba terhubung dengan jaringan *private blockchain* harus mendapatkan *permissions* oleh *Node* admin terlebih dahulu. Seperti yang terlihat dimana terdapat aksi yang dapat dilakukan *Node Client* dapat berupa *connect, send, receive*.

2. Pemberian Izin oleh *Node Admin*

```
multichain-cli grant 1Ef2MZGZBj7vC36g7bGtmXopDsAnLfpCjuHGeP
connect,send,receive
{"method":"grant","params":["1Ef2MZGZBj7vC36g7bGtmXopDsAnLfpCjuHGeP",
"connect, send, receive"],"id":"86226335-1705221510", "chain_name":"iotchain"}
0a4d46c91be3662e0491a85609e5ca932c7557cc16c72e153c599b2568ee5e6
```

Gambar 9. Pemberian Izin *Node Admin*

Gambar 9 merupakan proses pemberian izin oleh *node* admin kepada *node client* untuk melakukan koneksi *peer-to-peer* dan diikuti dengan perintah *grant* alamat dari *node client*, yang digunakan untuk memberikan *permission* agar *node client* dapat tergabung dalam jaringan *private blockchain* yang dikelola dan segala aksinya diatur oleh *node* admin.

Kemudian terdapat *public key* yang didapatkan dari pemberian *permission* oleh *node* admin dan menandakan proses koneksi *peer-to-peer* antara *node* admin dan *node client* telah berhasil.

3. *Publish Data*

```
multichain-cli iotchain publish iotdata key1
{"json":{"Suhu":"31C", "Kelembaban": "50%"},
{"method":"publish","params":["iotdata",
"key1",{"json":{"Suhu":"31C", "Kelembaban":"50%"}
}], "id": "29449075-1705221957", "chain_name":"iotchain"}
376a6f146c0eea975fd719f698bcef87f5c3ef83c73516f9bbf74893ee3a8fa6
```

Gambar 10. *Publish Data*

Gambar 10 merupakan proses *publish* data ke dalam *stream* yang dilakukan oleh *node* admin dengan menggunakan mode *write*, yang berarti dilakukan *input* data secara langsung dengan CLI pada terminal.

Kemudian didapatkan nilai *txid* dari proses *publish* data tersebut. Nilai *txid* diberikan sebagai bukti transaksi data pada setiap proses *publish* data ke dalam *stream* didalam *blockchain*.

4. Melihat Data

```

multichain-cli iotchain liststreamitems iotdata true 1
{"method":"liststreamitems","params":{"iotdata":true,1,"id":"28473826-1705222062","chain_name":"iotchain"}
[
  [
    {
      "publishers":["1Bv1dZvbDfowz4ctGKYjFAtj3acdsNQLxJNEkn"]
      "keys":["key1"]
      "offchain":false,
      "available":true,
      "data":{"json":{"suhu":"31c"},
      "Kelembaban":"50%",
      "confirmations":7,
      "blockhash":"008cfe63fe8334c8d864a82528753f4121b3906916e6e873af645726e06ce38d",
      "blockheight":44,
      "blockindex":1,
      "blocktime":1705221965,
      "txid":"376a6f146c0eaa975fd719f698bcef87f5c3ef83c73516f9bbf74893ee3a8fa6",

```

Gambar 11. Detail Data

Gambar 11 merupakan proses melihat data yang telah di *publish* sebelumnya pada *node* admin yang kemudian dijalankan dengan perintah *multichain-cli iotchain liststreamitems iotdata*.

Didapatkan detail item yang disimpan didalam *stream* dengan *publishers* adalah alamat dari *node* admin dan jumlah *confirmation* dari aksi *publish* data ditampilkan untuk mengathui jumlah validasi pada saat proses *publish* data berlangsung beserta nilai *txid* yang didapatkan sesuai dengan yang diberikan pada saat proses *publish* data.

5. Publish Data oleh Node Client

```

multichain-cli iotchain publish iotdata key2 '{"json":{"Suhu":"32C\\", "Kelembaban":"40%"}}'
{"method":"publish","params":{"iotdata","key2","json":{"Suhu":"32C","Kelembaban":"40%"}}, "id":"888031051705222168","chain_name":"iotchain"}
error code: -704
error message:
This wallet contains no addresses with permission to write to this stream and global send permission.

```

Gambar 12. Publish Data Node Client

Gambar 12 merupakan aksi *publish* data pada *node client* yang ditampilkan adalah *node client* tidak dapat menyimpan data pada *stream* karena belum memiliki *permission* untuk melakukan *publish* data ke dalam *stream*. *Error message* yang didapatkan menunjukkan bahwa alamat (*node client*) pada saat menyimpan data tidak terdapat alamat untuk disertakan sebagai bukti bahwa memiliki izin pada penyimpanan data pada *streams*.

6. Permissions Node Admin

```

multichain-cli iotchain grant 1Ef2MzGzBj7vC36g7bGtmXopDsAnLfpCjuHGep send
{"method":"grant","params":["1Ef2MzGzBj7vC36g7bGtmXopDsAnLfpCjuHGep","send"], "id":"34779544-170522223","chain_name":"iotchain"}
cfd98f9ce492ff718591a48abb33d1b368749dca0a3f36549b1d845c7aba146b
multichain-cli iotchain grant 1Ef2MzGzBj7vC36g7bGtmXopDsAnLfpCjuHGep iotdata.write
{"method":"grant","params":["1Ef2MzGzBj7vC36g7bGtmXopDsAnLfpCjuHGep","iotdata.write"], "id":"33022464-1705222260","chain_name":"iotchain"}
18ef9e88a7dd78b0a9423e502ef79fc34cfe1646cf2d9a0286a8d375a578bc2c

```

Gambar 13. Permissions Node Admin

Gambar 13 merupakan proses pemberian izin pada *node* admin kepada *node client* agar dapat melakukan *publish* data kedalam *stream* dengan dijalankan perintah *multichain-cli blockchain_name grant node_client_address send*, dimana diberi *permissions* agar *node client* dapat melakukan *send* data ke dalam *stream*.

Kemudian pada baris kedua dijalankan perintah *multichain-cli blockchain_name grant node_client_address iotdata.write* untuk *node client* dapat melakukan *publish* data dengan mode *write* ke dalam *stream* *iotdata*. Keduanya diikuti dengan nilai *hash* dari proses pemberian izin oleh *node* admin.

7. Melihat Data Node Client

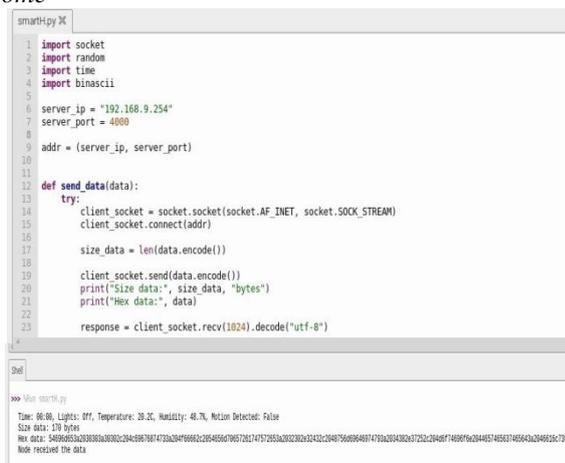
```
multichain-cli iotchain liststreamitems iotdata
{"method":"liststreamitems","params":["iotdata"],"id":"50650521-1705222323","chain_name":"iotchain"}
{
  "publishers" : ["1Ef2MZGZBj7vC36g7bGtmXopDsAnLfpCjuHGpP",
  "keys" : ["key2"
  "offchain" : false,
  "available" : true,
  "data" : {
    "json" : {
      "Suhu" : "32C",
      "Kelembaban" : "40%"
    }
  },
  "confirmations" : 3,
  "blocktime" : 1705222279,
  "txid" : "432650340bc94bce322b070d491368326a0772567d201b69f42782c6bde40dbf"
}
```

Gambar 14. Detail Data *Node Client*

Gambar 14 merupakan proses *subscribe* yang dilakukan oleh *node client* pada *stream node* admin dengan menjalankan perintah `multichain-cli blockchain_name subscribe iotdata` oleh *node client* agar dapat melihat detail item pada *stream*.

Kemudian dilakukan proses mencari detail *items* oleh *node client* dengan menjalankan perintah `multichain-cli blockchain_name liststreamitems iotdata` dan dengan kode *key2* yang merupakan hasil *publish* oleh *node client*. Terlihat *publishers* adalah alamat dari *node client* dan jumlah *confirmation* dari aksi *publish* data ditampilkan untuk mengathui jumlah validasi pada saat proses *publish* data berlangsung.

8. Pengiriman Data *Smart Home*



```
smartipy X
1 import socket
2 import random
3 import time
4 import binascii
5
6 server_ip = "192.168.9.254"
7 server_port = 4000
8
9 addr = (server_ip, server_port)
10
11
12 def send_data(data):
13     try:
14         client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
15         client_socket.connect(addr)
16
17         size_data = len(data.encode())
18
19         client_socket.send(data.encode())
20         print("Size data:", size_data, "bytes")
21         print("Hex data:", data)
22
23         response = client_socket.recv(1024).decode("utf-8")
24
25     except:
26         print("Error")
27
28 #
29
30 #
31
32 #
33
34 #
35
36 #
37
38 #
39
40 #
41
42 #
43
44 #
45
46 #
47
48 #
49
50 #
51
52 #
53
54 #
55
56 #
57
58 #
59
60 #
61
62 #
63
64 #
65
66 #
67
68 #
69
70 #
71
72 #
73
74 #
75
76 #
77
78 #
79
80 #
81
82 #
83
84 #
85
86 #
87
88 #
89
90 #
91
92 #
93
94 #
95
96 #
97
98 #
99
100 #
101
102 #
103
104 #
105
106 #
107
108 #
109
110 #
111
112 #
113
114 #
115
116 #
117
118 #
119
120 #
121
122 #
123
124 #
125
126 #
127
128 #
129
130 #
131
132 #
133
134 #
135
136 #
137
138 #
139
140 #
141
142 #
143
144 #
145
146 #
147
148 #
149
150 #
151
152 #
153
154 #
155
156 #
157
158 #
159
160 #
161
162 #
163
164 #
165
166 #
167
168 #
169
170 #
171
172 #
173
174 #
175
176 #
177
178 #
179
180 #
181
182 #
183
184 #
185
186 #
187
188 #
189
190 #
191
192 #
193
194 #
195
196 #
197
198 #
199
200 #
201
202 #
203
204 #
205
206 #
207
208 #
209
210 #
211
212 #
213
214 #
215
216 #
217
218 #
219
220 #
221
222 #
223
224 #
225
226 #
227
228 #
229
230 #
231
232 #
233
234 #
235
236 #
237
238 #
239
240 #
241
242 #
243
244 #
245
246 #
247
248 #
249
250 #
251
252 #
253
254 #
255
256 #
257
258 #
259
260 #
261
262 #
263
264 #
265
266 #
267
268 #
269
270 #
271
272 #
273
274 #
275
276 #
277
278 #
279
280 #
281
282 #
283
284 #
285
286 #
287
288 #
289
290 #
291
292 #
293
294 #
295
296 #
297
298 #
299
300 #
301
302 #
303
304 #
305
306 #
307
308 #
309
310 #
311
312 #
313
314 #
315
316 #
317
318 #
319
320 #
321
322 #
323
324 #
325
326 #
327
328 #
329
330 #
331
332 #
333
334 #
335
336 #
337
338 #
339
340 #
341
342 #
343
344 #
345
346 #
347
348 #
349
350 #
351
352 #
353
354 #
355
356 #
357
358 #
359
360 #
361
362 #
363
364 #
365
366 #
367
368 #
369
370 #
371
372 #
373
374 #
375
376 #
377
378 #
379
380 #
381
382 #
383
384 #
385
386 #
387
388 #
389
390 #
391
392 #
393
394 #
395
396 #
397
398 #
399
400 #
401
402 #
403
404 #
405
406 #
407
408 #
409
410 #
411
412 #
413
414 #
415
416 #
417
418 #
419
420 #
421
422 #
423
424 #
425
426 #
427
428 #
429
430 #
431
432 #
433
434 #
435
436 #
437
438 #
439
440 #
441
442 #
443
444 #
445
446 #
447
448 #
449
450 #
451
452 #
453
454 #
455
456 #
457
458 #
459
460 #
461
462 #
463
464 #
465
466 #
467
468 #
469
470 #
471
472 #
473
474 #
475
476 #
477
478 #
479
480 #
481
482 #
483
484 #
485
486 #
487
488 #
489
490 #
491
492 #
493
494 #
495
496 #
497
498 #
499
500 #
501
502 #
503
504 #
505
506 #
507
508 #
509
510 #
511
512 #
513
514 #
515
516 #
517
518 #
519
520 #
521
522 #
523
524 #
525
526 #
527
528 #
529
530 #
531
532 #
533
534 #
535
536 #
537
538 #
539
540 #
541
542 #
543
544 #
545
546 #
547
548 #
549
550 #
551
552 #
553
554 #
555
556 #
557
558 #
559
560 #
561
562 #
563
564 #
565
566 #
567
568 #
569
570 #
571
572 #
573
574 #
575
576 #
577
578 #
579
580 #
581
582 #
583
584 #
585
586 #
587
588 #
589
590 #
591
592 #
593
594 #
595
596 #
597
598 #
599
600 #
601
602 #
603
604 #
605
606 #
607
608 #
609
610 #
611
612 #
613
614 #
615
616 #
617
618 #
619
620 #
621
622 #
623
624 #
625
626 #
627
628 #
629
630 #
631
632 #
633
634 #
635
636 #
637
638 #
639
640 #
641
642 #
643
644 #
645
646 #
647
648 #
649
650 #
651
652 #
653
654 #
655
656 #
657
658 #
659
660 #
661
662 #
663
664 #
665
666 #
667
668 #
669
670 #
671
672 #
673
674 #
675
676 #
677
678 #
679
680 #
681
682 #
683
684 #
685
686 #
687
688 #
689
690 #
691
692 #
693
694 #
695
696 #
697
698 #
699
700 #
701
702 #
703
704 #
705
706 #
707
708 #
709
710 #
711
712 #
713
714 #
715
716 #
717
718 #
719
720 #
721
722 #
723
724 #
725
726 #
727
728 #
729
730 #
731
732 #
733
734 #
735
736 #
737
738 #
739
740 #
741
742 #
743
744 #
745
746 #
747
748 #
749
750 #
751
752 #
753
754 #
755
756 #
757
758 #
759
760 #
761
762 #
763
764 #
765
766 #
767
768 #
769
770 #
771
772 #
773
774 #
775
776 #
777
778 #
779
780 #
781
782 #
783
784 #
785
786 #
787
788 #
789
790 #
791
792 #
793
794 #
795
796 #
797
798 #
799
800 #
801
802 #
803
804 #
805
806 #
807
808 #
809
810 #
811
812 #
813
814 #
815
816 #
817
818 #
819
820 #
821
822 #
823
824 #
825
826 #
827
828 #
829
830 #
831
832 #
833
834 #
835
836 #
837
838 #
839
840 #
841
842 #
843
844 #
845
846 #
847
848 #
849
850 #
851
852 #
853
854 #
855
856 #
857
858 #
859
860 #
861
862 #
863
864 #
865
866 #
867
868 #
869
870 #
871
872 #
873
874 #
875
876 #
877
878 #
879
880 #
881
882 #
883
884 #
885
886 #
887
888 #
889
890 #
891
892 #
893
894 #
895
896 #
897
898 #
899
900 #
901
902 #
903
904 #
905
906 #
907
908 #
909
910 #
911
912 #
913
914 #
915
916 #
917
918 #
919
920 #
921
922 #
923
924 #
925
926 #
927
928 #
929
930 #
931
932 #
933
934 #
935
936 #
937
938 #
939
940 #
941
942 #
943
944 #
945
946 #
947
948 #
949
950 #
951
952 #
953
954 #
955
956 #
957
958 #
959
960 #
961
962 #
963
964 #
965
966 #
967
968 #
969
970 #
971
972 #
973
974 #
975
976 #
977
978 #
979
980 #
981
982 #
983
984 #
985
986 #
987
988 #
989
990 #
991
992 #
993
994 #
995
996 #
997
998 #
999
1000 #
1001
1002 #
1003
1004 #
1005
1006 #
1007
1008 #
1009
1010 #
1011
1012 #
1013
1014 #
1015
1016 #
1017
1018 #
1019
1020 #
1021
1022 #
1023
1024 #
1025
1026 #
1027
1028 #
1029
1030 #
1031
1032 #
1033
1034 #
1035
1036 #
1037
1038 #
1039
1040 #
1041
1042 #
1043
1044 #
1045
1046 #
1047
1048 #
1049
1050 #
1051
1052 #
1053
1054 #
1055
1056 #
1057
1058 #
1059
1060 #
1061
1062 #
1063
1064 #
1065
1066 #
1067
1068 #
1069
1070 #
1071
1072 #
1073
1074 #
1075
1076 #
1077
1078 #
1079
1080 #
1081
1082 #
1083
1084 #
1085
1086 #
1087
1088 #
1089
1090 #
1091
1092 #
1093
1094 #
1095
1096 #
1097
1098 #
1099
1100 #
1101
1102 #
1103
1104 #
1105
1106 #
1107
1108 #
1109
1110 #
1111
1112 #
1113
1114 #
1115
1116 #
1117
1118 #
1119
1120 #
1121
1122 #
1123
1124 #
1125
1126 #
1127
1128 #
1129
1130 #
1131
1132 #
1133
1134 #
1135
1136 #
1137
1138 #
1139
1140 #
1141
1142 #
1143
1144 #
1145
1146 #
1147
1148 #
1149
1150 #
1151
1152 #
1153
1154 #
1155
1156 #
1157
1158 #
1159
1160 #
1161
1162 #
1163
1164 #
1165
1166 #
1167
1168 #
1169
1170 #
1171
1172 #
1173
1174 #
1175
1176 #
1177
1178 #
1179
1180 #
1181
1182 #
1183
1184 #
1185
1186 #
1187
1188 #
1189
1190 #
1191
1192 #
1193
1194 #
1195
1196 #
1197
1198 #
1199
1200 #
1201
1202 #
1203
1204 #
1205
1206 #
1207
1208 #
1209
1210 #
1211
1212 #
1213
1214 #
1215
1216 #
1217
1218 #
1219
1220 #
1221
1222 #
1223
1224 #
1225
1226 #
1227
1228 #
1229
1230 #
1231
1232 #
1233
1234 #
1235
1236 #
1237
1238 #
1239
1240 #
1241
1242 #
1243
1244 #
1245
1246 #
1247
1248 #
1249
1250 #
1251
1252 #
1253
1254 #
1255
1256 #
1257
1258 #
1259
1260 #
1261
1262 #
1263
1264 #
1265
1266 #
1267
1268 #
1269
1270 #
1271
1272 #
1273
1274 #
1275
1276 #
1277
1278 #
1279
1280 #
1281
1282 #
1283
1284 #
1285
1286 #
1287
1288 #
1289
1290 #
1291
1292 #
1293
1294 #
1295
1296 #
1297
1298 #
1299
1300 #
1301
1302 #
1303
1304 #
1305
1306 #
1307
1308 #
1309
1310 #
1311
1312 #
1313
1314 #
1315
1316 #
1317
1318 #
1319
1320 #
1321
1322 #
1323
1324 #
1325
1326 #
1327
1328 #
1329
1330 #
1331
1332 #
1333
1334 #
1335
1336 #
1337
1338 #
1339
1340 #
1341
1342 #
1343
1344 #
1345
1346 #
1347
1348 #
1349
1350 #
1351
1352 #
1353
1354 #
1355
1356 #
1357
1358 #
1359
1360 #
1361
1362 #
1363
1364 #
1365
1366 #
1367
1368 #
1369
1370 #
1371
1372 #
1373
1374 #
1375
1376 #
1377
1378 #
1379
1380 #
1381
1382 #
1383
1384 #
1385
1386 #
1387
1388 #
1389
1390 #
1391
1392 #
1393
1394 #
1395
1396 #
1397
1398 #
1399
1400 #
1401
1402 #
1403
1404 #
1405
1406 #
1407
1408 #
1409
1410 #
1411
1412 #
1413
1414 #
1415
1416 #
1417
1418 #
1419
1420 #
1421
1422 #
1423
1424 #
1425
1426 #
1427
1428 #
1429
1430 #
1431
1432 #
1433
1434 #
1435
1436 #
1437
1438 #
1439
1440 #
1441
1442 #
1443
1444 #
1445
1446 #
1447
1448 #
1449
1450 #
1451
1452 #
1453
1454 #
1455
1456 #
1457
1458 #
1459
1460 #
1461
1462 #
1463
1464 #
1465
1466 #
1467
1468 #
1469
1470 #
1471
1472 #
1473
1474 #
1475
1476 #
1477
1478 #
1479
1480 #
1481
1482 #
1483
1484 #
1485
1486 #
1487
1488 #
1489
1490 #
1491
1492 #
1493
1494 #
1495
1496 #
1497
1498 #
1499
1500 #
1501
1502 #
1503
1504 #
1505
1506 #
1507
1508 #
1509
1510 #
1511
1512 #
1513
1514 #
1515
1516 #
1517
1518 #
1519
1520 #
1521
1522 #
1523
1524 #
1525
1526 #
1527
1528 #
1529
1530 #
1531
1532 #
1533
1534 #
1535
1536 #
1537
1538 #
1539
1540 #
1541
1542 #
1543
1544 #
1545
1546 #
1547
1548 #
1549
1550 #
1551
1552 #
1553
1554 #
1555
1556 #
1557
1558 #
1559
1560 #
1561
1562 #
1563
1564 #
1565
1566 #
1567
1568 #
1569
1570 #
1571
1572 #
1573
1574 #
1575
1576 #
1577
1578 #
1579
1580 #
1581
1582 #
1583
1584 #
1585
1586 #
1587
1588 #
1589
1590 #
1591
1592 #
1593
1594 #
1595
1596 #
1597
1598 #
1599
1600 #
1601
1602 #
1603
1604 #
1605
1606 #
1607
1608 #
1609
1610 #
1611
1612 #
1613
1614 #
1615
1616 #
1617
1618 #
1619
1620 #
1621
1622 #
1623
1624 #
1625
1626 #
1627
1628 #
1629
1630 #
1631
1632 #
1633
1634 #
1635
1636 #
1637
1638 #
1639
1640 #
1641
1642 #
1643
1644 #
1645
1646 #
1647
1648 #
1649
1650 #
1651
1652 #
1653
1654 #
1655
1656 #
1657
1658 #
1659
1660 #
1661
1662 #
1663
1664 #
1665
1666 #
1667
1668 #
1669
1670 #
1671
1672 #
1673
1674 #
1675
1676 #
1677
1678 #
1679
1680 #
1681
1682 #
1683
1684 #
1685
1686 #
1687
1688 #
1689
1690 #
1691
1692 #
1693
1694 #
1695
1696 #
1697
1698 #
1699
1700 #
1701
1702 #
1703
1704 #
1705
1706 #
1707
1708 #
1709
1710 #
1711
1712 #
1713
1714 #
1715
1716 #
1717
1718 #
1719
1720 #
1721
1722 #
1723
1724 #
1725
1726 #
1727
1728 #
1729
1730 #
1731
1732 #
1733
1734 #
1735
1736 #
1737
1738 #
1739
1740 #
1741
1742 #
1743
1744 #
1745
1746 #
1747
1748 #
1749
1750 #
1751
1752 #
1753
1754 #
1755
1756 #
1757
1758 #
1759
1760 #
1761
1762 #
1763
1764 #
1765
1766 #
1767
1768 #
1769
1770 #
1771
1772 #
1773
1774 #
1775
1776 #
1777
1778 #
1779
1780 #
1781
1782 #
1783
1784 #
1785
1786 #
1787
1788 #
1789
1790 #
1791
1792 #
1793
1794 #
1795
1796 #
1797
1798 #
1799
1800 #
1801
1802 #
1803
1804 #
1805
1806 #
1807
1808 #
1809
1810 #
1811
1812 #
1813
1814 #
1815
1816 #
1817
1818 #
1819
1820 #
1821
1822 #
1823
1824 #
1825
1826 #
1827
1828 #
1829
1830 #
1831
1832 #
1833
1834 #
1835
1836 #
1837
1838 #
1839
1840 #
1841
1842 #
1843
1844 #
1845
1846 #
1847
1848 #
1849
1850 #
1851
1852 #
1853
1854 #
1855
1856 #
1857
1858 #
1859
1860 #
1861
1862 #
1863
1864 #
1865
1866 #
1867
1868 #
1869
1870 #
1871
1872 #
1873
1874 #
1875
1876 #
1877
1878 #
1879
1880 #
1881
1882 #
1883
1884 #
1885
1886 #
1887
1888 #
1889
1890 #
1891
1892 #
1893
1894 #
1895
1896 #
1897
1898 #
1899
1900 #
1901
1902 #
1903
1904 #
1905
1906 #
1907
1908 #
1909
1910 #
1911
1912 #
1913
1914 #
1915
1916 #
1917
1918 #
1919
1920 #
1921
1922 #
1923
1924 #
1925
1926 #
1927
1928 #
1929
1930 #
1931
1932 #
1933
1934 #
1935
1936 #
1937
1938 #
1939
1940 #
1941
1942 #
1943
1944 #
1945
1946 #
1947
1948 #
1949
1950 #
1951
1952 #
1953
1954 #
1955
1956 #
1957
1958 #
1959
1960 #
1961
1962 #
1963
1964 #
1965
1966 #
1967
1968 #
1969
1970 #
1971
1972 #
1973
1974 #
1975
1976 #
1977
1978 #
1979
1980 #
1981
1982 #
1983
1984 #
1985
1986 #
1987
1988 #
1989
1990 #
1991
1992 #
1993
1994 #
1995
1996 #
1997
1998 #
1999
2000 #
2001
2002 #
2003
2004 #
2005
2006 #
2007
2008 #
2009
2010 #
2011
2012 #
2013
2014 #
2015
2016 #
2017
2018 #
2019
2020 #
2021
2022 #
2023
2024 #
2025
2026 #
2027
2028 #
2029
2030 #
2031
2032 #
2033
2034 #
2035
2036 #
2037
2038 #
2039
2040 #
2041
2042 #
2043
2044 #
2045
2046 #
2047
2048 #
2049
2050 #
2051
2052 #
2053
2054 #
2055
2056 #
2057
2058 #
2059
2060 #
2061
2062 #
2063
2064 #
2065
2066 #
2067
2068 #
2069
2070 #
2071
2072 #
2073
2074 #
2075
2076 #
2077
2078 #
2079
2080 #
2081
2082 #
2083
2084 #
2085
2086 #
2087
2088 #
2089
2090 #
2091
2092 #
2093
2094 #
2095
2096 #
2097
2098 #
2099
2100 #
2101
2102 #
2103
2104 #
2105
2106 #
2107
2108 #
2109
2110 #
2111
2112 #
2113
2114 #
2115
2116 #
2117
2118 #
2119
2120 #
2121
2122 #
2123
2124 #
2125
2126 #
2127
2128 #
2129
2130 #
2131
2132 #
2133
2134 #
2135
2136 #
2137
2138 #
2139
2140 #
2141
2142 #
2143
2144 #
2145
2146 #
2147
2148 #
2149
2150 #
2151
2152 #
2153
2154 #
2155
2156 #
2157
2158 #
2159
2160 #
2161
2162 #
2163
2164 #
2165
2166 #
2167
2168 #
2169
2170 #
2171
2172 #
2173
2174 #
2175
2176 #
2177
2178 #
2179
2180 #
2181
2182 #
2183
2184 #
2185
2186 #
2187
2188 #
2189
2190 #
2191
2192 #
2193
2194 #
2195
2196 #
2197
2198 #
2199
2200 #
2201
2202 #
2203
2204 #
2205
2206 #
2207
2208 #
2209
2210 #
2211
2212 #
2213
2214 #
2215
2216 #
2217
2218 #
2219
2220 #
2221
2222 #
2223
2224 #
2225
2226 #
2227
2228 #
2229
2230 #
2231
2232 #
2233
2234 #
2235
2236 #
2237
2238 #
2239
2240 #
2241
2242 #
2243
2244 #
2245
2246 #
2247
2248 #
2249
2250 #
2251
2252 #
2253
2254 #
2255
2256 #
2257
2258 #
2259
2260 #
2261
2262 #
2263
2264 #
2265
2266 #
2267
2268 #
2269
2270 #
2271
2272 #
2273
2274 #
2275
227
```

```

serChainy x  ServerChainy u  smartHome.py M  s.py  M  test.dat u
chainy: ~
1 import socket
2 from Savoir import Savoir
3
4 rpcuser = 'multichainrpc'
5 rpcpassword = 'Ebnw959kRzsd9MFR5xvndgZao4gg226800D0v2'
6 rpchost = '192.168.9.254'
7 rpcport = '4286'
8 chain_name = 'iotchain'
9
10 multichain = Savoir(rpcuser, rpcpassword, rpchost, rpcport, chain_name)
11
12 server_ip = '192.168.9.254'
13 server_port = 4800
14
15 addr = (server_ip, server_port)
16
17 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
18 server_socket.bind(addr)
19 server_socket.listen(5)
20 print(f'Server mendengarkan di {addr}')
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

jarkom@jarkom:~/iotchain$ python3 serChain.py
Server mendengarkan di ('192.168.9.254', 4800)
Menerima koneksi dari ('192.168.9.254', 44561)
Menerima data: 546966d653a2030303a30302c204c69676874733a204f666662c2054656d70657261747572653a2032302e32432c2048756d69646974793a2034382e37252c204d6f7469666e2044657465637465643a2046616c7365
Data telah disimpan dengan txid: ad0b015c1f88b478c1d86f65263834ea0f4d79b2b6fe32f9dc972f0f6ae67444

```

Gambar 16. Penerimaan Data *Smart Home*

Gambar 16 merupakan program *server* yang akan menerima koneksi socket dari klien untuk mendapat data hasil simulasi hasil program *Smart Home* kemudian akan dilakukan proses *publish* data ke dalam *stream* pada *node* admin.

Hasil dari program *server* yang menerima koneksi dengan program klien dan menerima data hasil simulasi dari program *Smart Home* yang nilai asli data telah di konversi kedalam format *hexadecimal*, kemudian akan di *publish* dalam *stream* pada *multichain* dengan diberi nilai *txid* data yang tersimpan ke dalam *stream* *multichain*.

10. Melihat Data *Smart Home*

```

jarkom@jarkom:~$ multichain-cli iotchain liststreamitems iotdata
false 1
{"method": "liststreamitems", "params": ["iotdata", false, 1], "id": "46942251-1713603276", "chain_name": "iotchain"}

"publishers": [
  "1Bv1dZvbDfowz4ctGRYjFAtj3acdsNQLxJNEkn"]

"keys": ["key3"]

"offchain": false,
"available": true,
"data": "546966d653a2030303a30302c204c69676874733a204f666662c2054656d70657261747572653a2032302e32432c2048756d69646974793a2034382e37252c204d6f7469666e2044657465637465643a2046616c7365",
"confirmations": 11,
"blocktime": 1713603132,
"txid": "ad0b015c1f88b478c1d86f65263834ea0f4d79b2b6fe32f9dc972f0f6ae67444"
}

```

Gambar 17. Detail Data dalam *Blockchain*

Gambar 17 merupakan pencarian data atau detail items hasil dari simulasi *Smart Home* yang disimpan melalui program *server* yang melakukan *publish* data ke dalam *stream* pada *multichain*. Nilai data yang sudah dikonversi ke dalam format *hexadecimal* dengan nilai *txid* yang didapatkan bernilai sama seperti pada Gambar 16 dan data diberikan juga keterangan *timestamp*.

6. Skenario Permissionless Device

1. Publish Data

```

multichain-cli iotchain publish iotdata key4
{"json": {"Suhu": "32C", "Kelembaban": "49%"}
error: Could not connect to the server 127.0.0.1:4286 (error code 1 - "EOF reached")

Make sure the multichain server is running and that you are connecting to the correct RPC port.

```

Gambar 18. Gagal Menyimpan Data

Gambar 18 merupakan aksi *Permissionless device* mencoba *publish* data pada *multichain* dengan menjalankan perintah *multichain-cli blockchain_name publish Streams_name key_value data_value* dan

didapatkan *error: Could not connect to the server* karena *permissionless device* tidak dapat terhubung dengan *server* multichain tetapi melakukan penyimpanan data.

2. Melihat Data

```
multichain-cli iotchain liststreamitems iotdata true 1
error: Could not connect to the server 127.0.0.1:4286 (error code 1
- "EOF reached")

Make sure the multichaind server is running and that you are
connecting to the correct RPC port.
```

Gambar 19. Gagal Melihat Data

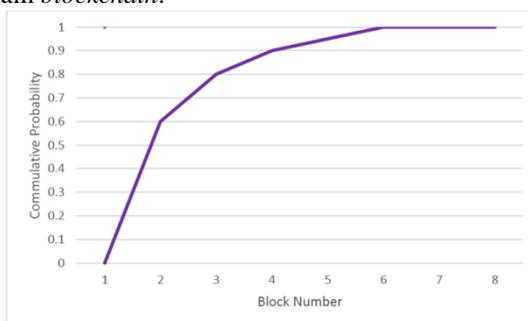
Gambar 19 merupakan aksi *Permissionless device* mencoba mendapatkan detail item dari percobaan *permissionless device* saat menyimpan data dengan menjalankan perintah *multichain-cli blockchain_name liststreamitems* dan didapatkan *error: Could not connect to the server* karena *permissionless device* tidak dapat terhubung dengan *server* multichain tetapi mencoba melakukan pencarian detail item.

7. Validasi Data

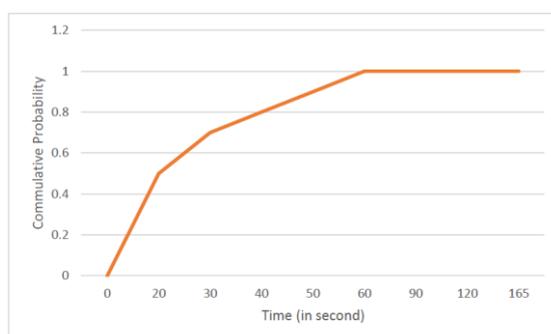
Validasi data mengacu pada proses pemberian izin *node client* untuk melakukan *mining* oleh *node* admin agar jaringan *private blockchain* yang berjalan pada sisi *node client* tidak mengalami pembekuan atau data dan jumlah *block* pada kedua *node* tidak sinkron serta tidak dapat lagi menyimpan data ke dalam stream pada jaringan *private blockchain*. Proses validasi didasarkan pada nilai *spacing* yang harus dipenuhi dengan mengkombinasikan nilai parameter *mining-diversity* dan jumlah *miners* atau nilai $spacing = mining-diversity * miners$.

8. Evaluasi Sistem

Evaluasi kinerja dilakukan untuk dapat mengetahui nilai yang dapat dihasilkan oleh *blockchain* yang diimplementasikan pada proses distribusi data IoT kedalam jaringan *private blockchain* dengan menggunakan platform multichain dan didapatkan nilai kemungkinan kumulatif berdasarkan jumlah *block number* yang dihasilkan didalam *blockchain*.



Gambar 20. *Commulative Probability Terhadap Block Number*



Gambar 21. *Commulative Probability Terhadap Time*

Nilai kemungkinan kumulatif (*commulative probability*) berdasarkan jumlah *block* pada Gambar 20 tampak dengan nilai kemungkinan kumulatif (*commulative probability*) dihasilkan dari nilai *block* yang bertambah setiap 15 detik sampai mendapatkan jumlah 6 konfirmasi dari prinsip *blockchain* sehingga suatu *block* dinyatakan valid dalam jaringan *blockchain*, yang mengartikan setelah *block* ditambahkan ke rantai dibutuhkan 6 *block* tambahan untuk memastikan keamanan transaksi. Kemudian didapatkan nilai kemungkinan kumulatif (*commulative probability*) berdasarkan jumlah waktu pada Gambar 21 tampak yang dibutuhkan membuat *block* didalam *blockchain*, dengan nilai kemungkinan kumulatif dihasilkan dari nilai *block* yang bertambah setiap 15 detik sampai mendapatkan jumlah 6 konfirmasi dari prinsip *blockchain* sehingga suatu *block* dinyatakan valid dalam jaringan *blockchain*. Kemudian hingga suatu *block* dinyatakan valid saat waktu mencapai 60 detik sampai 165 detik.

4. CONCLUSION

Implementasi *blockchain* sebagai sistem keamanan IoT dalam mendistribusikan dan menjaga privasi data mampu memberikan hasil pengamanan data berupa *block* yang berisi *hash block*, *miner*, *confirmation*, *previous block hash* dan *next block hash* serta *timestamp* dan masing-masing *block* saling terhubung. Hal ini membuktikan bahwa implementasi *blockchain* mampu menjaga keamanan dan privasi data pada saat melalui proses distribusi oleh perangkat IoT hingga sampai pada akhirnya data disimpan. Dari hasil pengujian yang dilakukan waktu dan pembuatan *block* yang membutuhkan waktu 15 detik dalam setiap pembuatan *block* serta dibersamai dengan proses validasi yang dilakukan saat membuat *block* hingga mendapatkan jumlah 6 konfirmasi yang sesuai dengan prinsip *blockchain* sehingga *block* dinyatakan valid dalam jaringan *blockchain*.

Acknowledgments

Diucapkan terimakasih pada Laboratorium Teknik Komputer Jurusan Teknik Elektro Universitas Mataram atas peminjaman perangkat penelitian.

2. REFERENCES (10 PT)

- [1] S. R. Lam, S. Jain, and R. Doriya, (2021). Security threats and solutions to IoT using *Blockchain*: A review. Proceedings – 5th Internasional Conference on Intelligent Computing and Control System, ICICCS 2021.
- [2] A. Amrulloh, E. Ujianto, (2019), *Kripto Simetris Menggunakan Algoritma Vigenere Cipher*. Jurnal CoreIT, Vol. 5, No. 2
- [3] K. Andersson, LCN Symposium 2019 : 2019 IEEE 44th Local Computer Networks Symposium on Emerging Topics in Networking : Proceedings : 14 October 2019, Osnabrück, Germany.
- [4] D. A. Badawi, 2019. "Investigasi Forensik Digital Berbasis Teknologi *Blockchain*." Universitas Islam Indonesia. [https://dspace.uui.ac.id/handle/123456789/16982%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/16982/08.naskah publikasi .pdf?sequence=11&isAllowed=y](https://dspace.uui.ac.id/handle/123456789/16982%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/16982/08.naskah%20publikasi.pdf?sequence=11&isAllowed=y).
- [5] A. Dorri, S.S Kanhere, R. Jurdak, P. Gauravaram, (2017). *Blockchain* for IoT security dan privacy: The case study of a *Smart Home*, IEEE International Conference on Pervasive Computing and Communications 15. 2017 Kona, Hawaii et al. 2017 IEEE International Conference on Pervasive Computing and Communications.
- [6] Y. Efendi,. 2018. "*Internet of Things (Iot)* Sistem Pengendalian Lampu Menggunakan *Raspberry Pi* Berbasis Mobile." Jurnal Ilmiah Ilmu Komputer 4(2): 21–27.
- [7] E. Fernando, Meyliana, and Surjandy. 2019. "*Blockchain* Technology Implementation in *Raspberry Pi* for *Private* Network." Proceedings of 2019 4th International Conference on Sustainable Information Engineering and Technology, SIET 2019: 154–58.
- [8] R. Hargude, R., Ashutosh, Ghule., A. Nawale, S. Adshure, (2021), "Verification and Validation of Certificate Using *Blockchain*". International Journal of Engineering Research & Technology, Departemen Komputer India, Vol.6

- [9] A. Hayes, (2023). *Blockchain* Fact: What is it, how it works, and how it can be used. [Online]. Available: <https://www.investopedia.com/terms/b/blockchain.asp#what-is-blockchain>
- [10] Husni., (2019), *Mengenal Blockchain: Teknologi dibelakang Bitcoin*. WordPress. Tersedia di : <https://komputasi.files.wordpress.com/2019/05/husni-mengenal-blockchain-teknologi-di-belakang-bitcoin.pdf>, diakses 18 April 2023.
- [11] J. Frankenfield, 2019, “Proof of Work”: Investopedia, 28 June 2020. [online]. Available: <https://www.investopedia.com/terms/p/proofwork.asp#:~:text=Proof%20of%20work%20describes%20a,launching%20denial%20of%20service%20attacks>.
- [12] J. Frankenfield, 2019, “ Proof of Stake” Investopedia, 11 Agustus 2019. [online]. Available: <https://www.investopedia.com/terms/p/proof-of-stake-pos.asp>.
- [13] M.A. Khan, K. Salah, (2018). IoT Security: Review, *blockchain* solution, and open challenges. Future generation computer system.
- [14] R. V. Nesterenko dan M. A. Maslova (2022). Menggunakan Teknologi *Blockchain* Untuk Memastikan Keamanan. Jurnal Nasional Teknologi Komputer Vol. 2: No:1 ; Januari 2022. E-ISSN: 2808-4845; P-ISSN: 2808-7801.
- [15] S. Pahlajani, K. Avinash, and P. Vinod, “Algorithms.” 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT): 1–6.
- [16] A. Panarello, N. Tapas, G. Merlino F. Longo, A. Puliafito, 2018, *Blockchain* and IoT integration: A systematic survey. Sensors (Switzerland).
- [17] R. Premkumar, and S. P. Sathya. 2021. “A *Blockchain* Based Framework for IoT Security.” In Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021, Institute of Electrical and Electronics Engineers Inc., 409–13.
- [18] Rumah. (2023). [online]. Apa itu Multichain di *Blockchain*. Artikel rumah. Available: https://id.tishijie.com/10609/Apa_itu_MultiChain_di_blockchain.
- [19] S. Seth, (2022) Explained Crypto in Cryptocurrency. [online]. Available: <https://www.investopedia.com/tech/explaining-crypto-cryptocurrency/>
- [20] U. Ikram, 2019. “The Internet of Things: A Review of Enabled Technologies and Future Challenges.” IEEE Access 7: 7606–40.
- [21] T. P. Utomo, 2022. “Implementasi Teknologi *Blockchain* Di Perpustakaan: Peluang, Tantangan Dan Hambatan.” Buletin Perpustakaan 4(2): 173–200.
- [22] I. D. Wijaya, N. Usman., and M. A. Barata, (2018). “Implementasi *Raspberry Pi* Untuk Rancang Bangun Sistem Keamanan Pintu Ruang *Server* Dengan Pengenalan Wajah Menggunakan Metode *Triangle Face*”.