

MAKALAH SEMINAR HASIL TUGAS AKHIR

**IMPLEMENTASI *BLOCKCHAIN* SEBAGAI SISTEM
KEAMANAN IoT DALAM MENDISTRIBUSIKAN DAN
MENJAGA PRIVASI DATA**



Oleh :

Lalu Ocky Saktiya Luhung

F1B 019 079

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2023

Makalah Seminar Hasil Tugas Akhir

**IMPLEMENTASI *BLOCKCHAIN* SEBAGAI SISTEM
KEAMANAN IoT DALAM MENDISTRIBUSIKAN DAN
MENJAGA PRIVASI DATA**

Oleh:

**Lalu Ocky Saktiya Luhung
F1B 019 079**

Telah diperiksa dan disetujui oleh:

1. Pembimbing Utama

Dr. Misbahuddin, S.T., M.T. IPU.
NIP: 19681005 199703 1 001

Tanggal:

2. Pembimbing Pendamping

Giri Wahyu Wiriasto, S.T., M.T.
NIP: 19820904 201012 1 001

Tanggal:

Mengetahui
Ketua Jurusan/Prodi Teknik ...
Fakultas Teknik
Universitas Mataram

A. Sjamsjiar Rachman, ST., MT
NIP : 19711124 199903 1 004

DAFTAR ISI

MAKALAH SEMINAR HASIL TUGAS AKHIR	i
Makalah Seminar Hasil Tugas Akhir	ii
Fakultas Teknik	ii
Universitas Mataram	ii
DAFTAR ISI	iii
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
ABSTRAK	viii
<i>ABSTRACT</i>	ix
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Sistematika Penulisan	4
BAB II	6
TINJAUAN PUSTAKA DAN DASAR TEORI	6
2.1. Tinjauan Pustaka	6
2.2. Dasar Teori	7
2.2.1 <i>Internet of Things (IoT)</i>	7
2.2.2 <i>Blockchain</i>	8
2.2.3 <i>Blockchain-IoT</i>	10
2.2.4 <i>Algoritma Consensus</i>	11
2.2.5 <i>Kriptografi</i>	12
2.2.6 <i>MultiChain</i>	13
2.2.7 <i>Socket Programming</i>	13
BAB III	14
METODE PENELITIAN	14

3.1.	Tahapan Penelitian	14
3.2.	Studi Literatur.....	15
3.3.	Analisis Kebutuhan	15
3.4.	Perancangan Topologi	15
3.4.1	Topologi Jaringan.....	15
3.4.2	Alur Komunikasi <i>Blockchain</i>	17
3.4.3	Perangkat <i>Smart Home</i>	17
3.4.4	<i>Node admin</i>	18
3.4.5	Pengiriman Data	19
3.4.6	Penerimaan Data	19
3.5.	Pengujian	21
3.5.1	Koneksi dengan MultiChain	21
3.5.2	Penyimpanan Data	21
3.5.3	Validasi Data	21
BAB IV		22
HASIL DAN PEMBAHASAN		22
4.1.	Implementasi Multichain.....	22
4.1.1.	Proses <i>Create Blockchain</i>	22
4.1.2.	Proses <i>Run Blockchain</i>	22
4.1.3.	Proses Penelusuran <i>Blockchain</i>	23
4.1.4.	Proses Mencari <i>Permissions</i>	25
4.2.	Koneksi <i>Node client</i> dengan <i>Node admin</i>	26
4.2.1.	Proses Koneksi oleh <i>Node Client</i>	26
4.2.2.	Proses <i>Permissions</i> oleh <i>Node admin</i>	26
4.2.3.	Proses <i>Run</i> pada <i>Node Client</i>	27
4.2.4.	Proses Penelusuran pada <i>Node Client</i>	27
4.3.	Membuat <i>Stream</i>	30
4.3.1.	<i>Create Stream</i>	30
4.3.2.	Mencari <i>List Stream</i>	30
4.3.3.	Meyimpan Data pada <i>Stream</i>	31
4.3.4.	Melihat Detail Data pada <i>Stream</i>	31
4.4.	Penyimpanan Data <i>Node Client</i> pada <i>Stream</i>	32

4.4.1.	<i>Node client</i> Mencari Detail Item pada <i>Stream</i>	32
4.4.2.	<i>Node client</i> Gagal Menyimpan Data pada <i>Stream</i>	33
4.4.3.	<i>Permission</i> oleh <i>Node admin</i>	33
4.4.4.	<i>Node Client</i> Berhasil Meyimpan Data pada <i>Stream</i>	33
4.4.5.	Pencarian Detail Item pada <i>Stream</i>	34
4.5.	Program Perangkat <i>Smart Home</i>	35
4.6.	Program <i>Server</i> Untuk Terhubung dengan Multichain	36
4.7.	Pencarian Data pada <i>Stream</i>	38
4.8.	<i>Permissionless Device</i>	38
4.8.1.	<i>Permissionless Device</i> Mencoba Terhubung	39
4.8.2.	<i>Permissionless Device</i> Menyimpan Data.....	39
4.8.3.	<i>Permissionless Device</i> Mencari Detail Items.....	39
BAB V		41
KESIMPULAN DAN SARAN		41
5.1.	Kesimpulan.....	41
5.2.	Saran	41
DAFTAR PUSTAKA		43

DAFTAR TABEL

DAFTAR GAMBAR

Gambar 2.1 <i>Blockchain</i> dan Database	9
Gambar 2.2 Struktur <i>Blockchain</i>	10
Gambar 3.1 Diagram Alir Tahapan Penelitian	14
Gambar 3.2 Topologi Jaringan	16
Gambar 3.3 Alur Komunikasi <i>Blockchain</i>	17
Gambar 3.4 Diagram Alir Program Simulasi Smart Home	18
Gambar 3.5 Alur Penerimaan Data	20
Gambar 4.1 Create Private blockchain	22
Gambar 4.2 Running Blockchain	23
Gambar 4.3 Mendapatkan Info Blockchain	24
Gambar 4.4 Mendapatkan Detail Block	25
Gambar 4.5 Mencari List Permission dalam Blockchain	25
Gambar 4. 6 Node client Mencoba Terhubung dengan Jaringan Private blockchain	26
Gambar 4.7 Pemberian Izin untuk Terhubung oleh Node admin	26
Gambar 4.8 Node client Berhasil Terhubung	27
Gambar 4.9 Mendapatkan Detail Blocks pada Node Client	27
Gambar 4.10 Peer info pada Node admin	28
Gambar 4.11 Peer Info pada Node Client	29
Gambar 4.12 Create Stream	30
Gambar 4.13 Mencari List Stream	30
Gambar 4.14 Publish Data pada Stream	31
Gambar 4.15 Gagal Melihat Detail Items	31
Gambar 4.16 Proses Subscribe Stream	31
Gambar 4.17 Detail Items dalam Stream	32
Gambar 4.18 Node client Gagal Mendapat Detail Items	32
Gambar 4.19 Node client Gagal Menyimpan Data	33
Gambar 4.20 Pemberian Izin oleh Node admin	33
Gambar 4.21 Node client Berhasil Meyimpan Data	33
Gambar 4.22 Proses Subscribe dan Mendapatkan Detail Items	34
Gambar 4.23 Hasil Simulasi Program Smart Home	36
Gambar 4. 24 Menerima dan Publish Data oleh Server	38
Gambar 4.25 Details Items dari Data Smart Home	38
Gambar 4.26 Permissionless Device Mencoba Terhubung	39
Gambar 4.27 Permissionless Device Gagal Menyimpan Data	39
Gambar 4.28 Permissionless Device Gagal Mencari Detail Item	39

ABSTRAK

Integrasi teknologi *blockchain* dengan *Internet of Things* (IoT) untuk meningkatkan keamanan dan privasi data. Dengan fokus pada keamanan IoT pada 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. 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.

Kata Kunci : *Blockchain, Internet of Things (IoT), Private blockchain*

ABSTRACT

Integration of blockchain technology with the Internet of Things (IoT) to improve data security and privacy. By focusing on IoT security when distributing data, the risk of data manipulation can be reduced because blockchain enables a decentralized and encrypted structure. Data integrity is enhanced through transaction verification and data storage authorized by hash value and proof of transaction (txid). The use of a private blockchain network controls data access, enhances security, privacy, and ensures access is limited to authorized parties. The results show that blockchain can improve IoT data security and better maintain data integrity, strengthen authorization and control data access more selectively.

Keyword : *Blockchain, Internet of Things (IoT), Private blockchain*

BAB I

PENDAHULUAN

1.1.Latar Belakang

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 (Nesterenko dan Maslova, 2022).

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*. 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 *hashing* 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 berperilaku korup di masa depan, meskipun mereka dapat dipercaya sekarang (Khan, dkk, 2018).

1.2.Rumusan Masalah

Berdasarkan latar belakang diatas dapat diidentifikasi beberapa permasalahan yaitu :

1. Bagaimana cara menjaga keamanan data dalam sistem IoT dengan menggunakan *blockchain*?
2. Bagaimana *blockchain* dapat diimplemetasikan pada sistem keamanan IoT untuk meningkatkan keamanan dan privasi data pada saat mendistribusikan data?
3. Bagaimana *blockchain* dapat memfasilitasi proses autentikasi dan otorisasi pada sistem keamanan IoT?

1.3.Batasan Masalah

Penelitian yang dilakukan memiliki batasan masalah dan ruang lingkup sebagai berikut:

1. Menjaga keamanan dan privasi data dalam sistem IoT berfokus dengan menggunakan jaringan *private blockchain*.
2. Perancangan dan pengembangan sistem keamanan IoT berbasis *blockchain*.
3. Analisis keamanan dan privasi data saat mendistribusikan data oleh perangkat IoT kepada *blockchain*.

1.4.Tujuan Penelitian

Tujuan penelitian ini adalah mengimplementasi dan mengintegrasikan teknologi IoT dengan teknologi *blockchain* dalam mendistribusikan data pada IoT dengan menggunakan *blockchain*, dan berbagai hal lainnya seperti berikut:

1. Menjaga keamanan dan privasi data pada sistem IoT dengan *blockchain*.
2. Meningkatkan integritas data, memberikan otentikasi dan otorisasi yang aman, memfasilitasi transaksi yang aman, serta meningkatkan privasi data.
3. Menggunakan jaringan *private blockchain* dalam memberi akses secara khusus.

1.5. Manfaat Penelitian

Manfaat dari hasil pembuatan Tugas Akhir ini adalah :

1. Dapat meningkatkan keamanan sistem IoT yang sudah ada dengan mengintegrasikan dengan teknologi *blockchain*.
2. Mendapatkan pemahaman implementasi *blockchain* pada sistem keamanan IoT.
3. Menggunakan dan mengintegrasikan IoT dengan *Blockchain* dapat meningkatkan keamanan dan privasi data.

1.6. Sistematika Penulisan

Adapun sistematika penulisan dari Laporan Tugas Akhir ini terbagi menjadi beberapa bab, sesuai dengan ketentuan yang ditetapkan di Fakultas Teknik, Universitas Mataram sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi gambaran umum dan penjelasan mengenai latar belakang pemilihan judul penelitian, perumusan masalah yang dimana merupakan tulisan singkat berisi pertanyaan tentang topik penelitian yang nantinya akan dijawab oleh penulis sehingga penelitian yang dilakukan memiliki suatu kesimpulan dari hasil analisis dan visualisasi data penelitian yang dilakukan, selain itu terdapat pula batasan masalah yang merupakan batas-batas dari topik penelitian yang sedang dikaji atau diteliti, serta terdapat juga tujuan penelitian dan manfaat penelitian yang merupakan keuntungan yang didapat atau diperoleh oleh berbagai pihak dari penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini berisi mengenai tinjauan pustaka dan landasan teori, yang dimana dalam tinjauan pustaka mengulas beberapa penelitian sebelumnya yang sejenis mengenai *blockchain* untuk sistem keamanan pada IoT, sedangkan pada dasar teori membahas mengenai teori-teori yang berkaitan dengan *blockchain*

untuk sistem keamanan pada IoT yang akan dilakukan pada penelitian yang didapatkan melalui beberapa sumber-sumber seperti jurnal, buku, dsb.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan tentang langkah-langkah dalam pelaksanaan penelitian yang dimana terdiri dari alur penelitian yang membahas mengenai gambaran umum dalam bentuk diagram terkait alur penelitian yang dilakukan dan uraian metodologi yang membahas tahapan yang dilakukan pada alur penelitian diantaranya yaitu, pengambilan data, pengiriman data, pembuatan jaringan *blockchain* pada sistem IoT, penerimaan data, validasi dan otentikasi keamanan data dalam *blockchain*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil dari penelitian yang telah dilakukan yang dimana terdiri dari hasil analisa, perancangan jaringan *blockchain*, dan pengujian pada perangkat IoT yang menyimpan data pada jaringan *private blockchain*.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang merupakan jawaban dari masalah yang dirumuskan dalam bentuk kalimat tanya di rumusan masalah dan berisi saran yang membangun untuk pengembangan yang lebih baik di masa depan.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

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 adalah "*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 (Dorri dkk., 2017).

Penelitian kedua adalah "*A secure and private data sharing architecture for IoT based on blockchain and smart contract*". Penelitian ini bertujuan untuk memperkenalkan arsitektur berbagi data yang aman dan pribadi untuk *Internet of Things* (IoT) menggunakan *blockchain* dan *smart contract*. Penelitian ini mengusulkan arsitektur berbagi data yang terdiri dari beberapa komponen, yaitu penyedia layanan, pengguna, dan *blockchain*. Dalam arsitektur ini, *blockchain* digunakan untuk memastikan keamanan dan privasi data, sedangkan *smart contract* digunakan untuk mengatur dan mengelola data yang dibagikan. Hasil dari penelitian ini menunjukkan bahwa arsitektur berbagi data dapat memberikan

tingkat keamanan dan privasi yang lebih tinggi dibandingkan dengan arsitektur yang tidak menggunakan *blockchain*. Selain itu, arsitektur yang diusulkan juga lebih efisien dalam hal penggunaan sumber daya dan waktu respons (Mingyu, dkk.,2020).

Penelitian ketiga adalah "*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 (Alam dkk., 2021).

2.2. Dasar Teori

2.2.1 *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. *Internet of Things* atau sering disebut IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. 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 (Efendi, 2018).

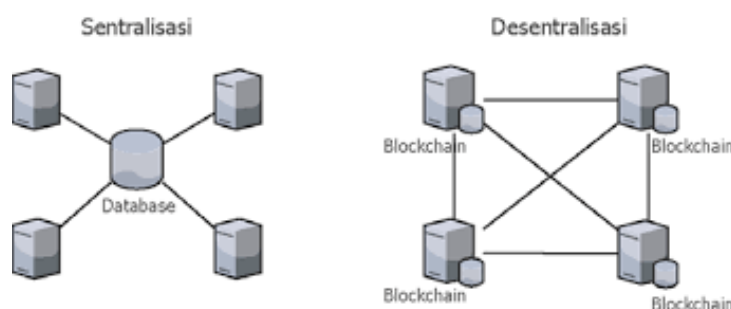
Dengan meningkatnya penggunaan perangkat pintar, kemampuan bertahan dan pengorganisasian mandiri jaringan menjadi sangat menantang. Namun, beberapa paradigma pengorganisasian mandiri yang diusulkan, dapat meningkatkan kekuatan jaringan. Dalam IoT, asosiasi dari berbagai perangkat heterogen mengurangi kemampuan jaringan sumber daya, yang menarik perhatian para peneliti ke arah bidang yang sedang berkembang ini. Perkembangan IoT mengarah pada sejumlah besar pembuatan konten, yang memperoleh unit pemrosesan besar, penyimpanan konten (*cache*), dan penyediaan *bandwidth*. Hal ini disebabkan karena fakta bahwa jumlah *node* kecil yang terhubung ke Internet akan mencapai 27 miliar pada tahun 2021. 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 (Ikram dkk., 2019).

2.2.2 Blockchain

Sejak Bitcoin diperkenalkan pada tahun 2009, penggunaan *blockchain* telah meledak melalui penciptaan berbagai mata uang kripto, *decentralized finance* (DeFi) *applications*, *non-fungible tokens* (NFT), dan *smart contract*.

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.

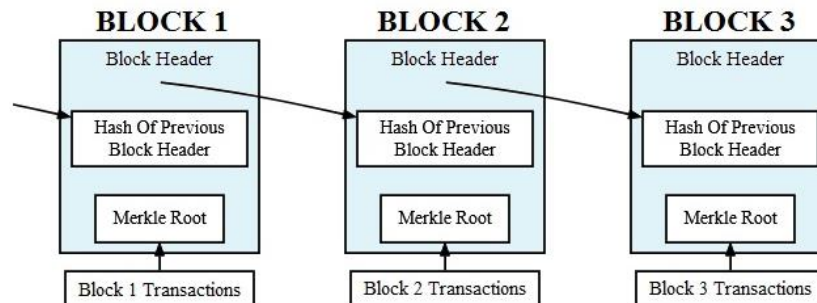


Gambar 2.1 *Blockchain dan Database*

Blockchain terdiri dari program yang disebut skrip yang melakukan tugas-tugas yang biasanya dilakukan dalam *database*: Memasukkan dan mengakses informasi serta menyimpan dan menyimpannya di suatu tempat. *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. (Hayes,2023).

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 memungkinkan 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 dibawah ini (Premkumar and Sathya 2021).



Gambar 2.2 Struktur *Blockchain*

2.2.3 *Blockchain-IoT*

Blockchain menyediakan penyimpanan data yang terdesentralisasi untuk sistem Teknologi Informasi (TI) umum. *Blockchain* diperkenalkan pada akhir tahun 2008 dan berfungsi sebagai platform untuk dan pemrosesan transaksi anonim menggunakan desentralisasi terdesentralisasi menggunakan jaringan komputer biasa. Biasanya, *Blockchain* dibangun sebagai sebuah daftar yang terhubung dari blok data, di mana mengubah satu bit di salah satu blok yang disimpan sebelumnya dapat segera ditemukan oleh rekan-rekan yang berpartisipasi. Dalam *Blockchain*, penambang adalah aktor utama utama yang memverifikasi validitas data yang disimpan. Kasus penggunaan yang terintegrasi dengan IoT telah meningkatkan perhatian yang tinggi dalam dekade terakhir, karena pemantauan rantai pasokan, pemantauan lingkungan, kota pintar, industri pintar, dan perawatan kesehatan berfokus pada kekekalan data dan membutuhkan sistem IoT untuk pengukuran, pengumpulan data, dan kontrol aktif. Dengan demikian, maka integrasi *Blockchain* dan IoT dapat mendukung BIoT menanggapi tuntutan penyimpanan yang persisten dan data sangat aman, di mana pengumpulan data otomatis menjadi kunci untuk menawarkan transparansi dan keandalan. BIoT yang sangat menuntut membutuhkan elaborasi dan analisis protokol IoT yang mendasari, yang membentuk dasar komunikasi untuk sistem IoT. Dengan demikian, studi tentang jangkauan komunikasi, kecepatan data, *maximum transmission units* (MTU), keandalan protokol komunikasi dan

efisiensi energi diperlukan untuk mendukung dengan tepat Penerapan IoT (Andersson dkk., 2019).

2.2.4 Algoritma *Consensus*

Algoritma *consensus* adalah sebuah teknik untuk mencapai sebuah kesepakatan bersama di dalam sebuah kelompok. Seperti sebuah kelompok yang terdiri dari sepuluh *node* yang perlu menentukan pilihan tentang tugas yang menguntungkan mereka semua. Semuanya dapat mengusulkan sebuah pemikiran, namun yang dominan akan disetujui oleh salah satu yang paling mendorong mereka. Namun, perhitungan konsensus tidak hanya setuju dengan bagian yang dominan memberikan suara, namun juga menyetujui yang menguntungkan semua pihak. 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* (Pahlajani, Kshirsagar, and Pachghare, 2019). *Blockchain* menerapkan beberapa algoritma *consensus* seperti berikut :

1. *Proof of Work* (PoW)

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 disebarkan 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 (Frankenfield, 2019).

2. *Proof of Stake* (PoS)

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 (Frankenfield, 2019).

2.2.5 Kriptografi

Teknologi kriptografi adalah kemampuan untuk bertukar pesan yang hanya dapat dibaca oleh penerima yang dituju dan digunakan untuk beberapa tujuan untuk mengamankan berbagai transaksi yang terjadi di jaringan, untuk mengontrol pembuatan unit mata uang baru, dan untuk verifikasi transfer aset digital dan token. Mata uang digital meniru konsep tanda tangan dunia nyata dengan menggunakan teknik kriptografi dan kunci enkripsi. Metode kriptografi menggunakan kode matematika tingkat lanjut untuk menyimpan dan mengirimkan nilai data dalam format yang aman yang memastikan hanya mereka yang memiliki data atau transaksi tersebut yang dapat menerima, membaca, dan memproses data, serta memastikan keaslian transaksi dan partisipan, seperti halnya tanda tangan di dunia nyata.

Pada teknologi *blockchain* kriptografi menggunakan metode *Hashing*, yang digunakan untuk memverifikasi integritas data transaksi di jaringan secara efisien. *Hashing* menjaga struktur data *blockchain*, mengkodekan alamat akun

seseorang, merupakan bagian integral dari proses enkripsi transaksi yang terjadi antar akun, dan memungkinkan penambahan blok. Selain itu, tanda tangan digital melengkapi berbagai proses kriptografi ini, dengan mengizinkan partisipan asli untuk membuktikan identitas mereka ke jaringan (Seth, 2022).

2.2.6 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 penambahan (*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 (Rumah, 2023).

2.2.7 Socket Programming

Socket adalah sebuah cara untuk berkomunikasi dengan program atau *node* lain dengan menggunakan komunikasi antara *client* dan *server*. Agar suatu *socket* dapat berkomunikasi dengan *socket* lainnya, maka *socket* butuh diberi suatu alamat untuk identifikasi. Alamat *socket* terdiri dari alamat IP dan nomer *port*. Alamat IP dapat menggunakan alamat jaringan lokal (LAN) maupun internet.

BAB III

METODE PENELITIAN

3.1. Tahapan Penelitian

Perancangan alur dari tahapan penelitian ini merupakan gambaran umum dari tahapan penelitian yang akan dilakukan dalam pengerjaan tugas akhir mulai dari awal hingga akhir. Tahapan penelitian tugas akhir digambarkan seperti berikut:



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.2. Studi Literatur

Pada penelitian ini dilakukan studi literatur sebagai tahap awalan untuk melakukan penelitian, tujuan dari studi literatur ini adalah untuk memahami dasar teori dan konsep yang akan menjadi pendukung dalam melakukan penelitian yang berjudul "Implementasi *Blockchain* Sebagai Sistem Keamanan IoT dalam Mendistribusikan dan Menjaga Privasi Data".

3.3. Analisis Kebutuhan

1. *Hardware*

Laptop HP Notebook 14-am013TU. Intel® Celeron® CPU N3060@ 1.60GHz with 8.0GB RAM

2. *Software*

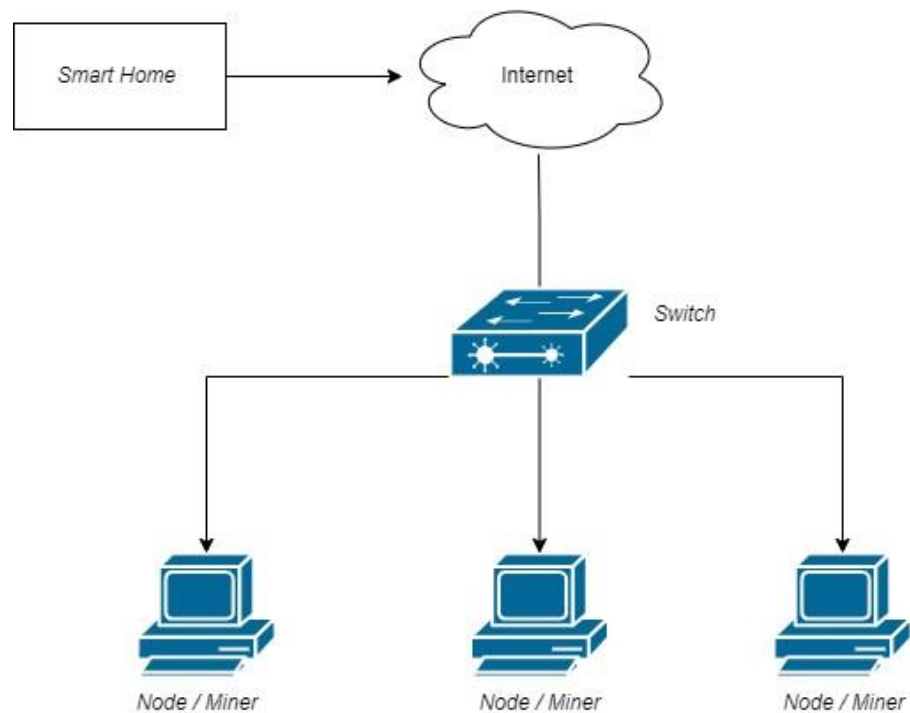
Software yang digunakan pada penelitian ini sebagai berikut:

- a. Visual Studio Code sebagai *text editor* yang membantu proses develop program simulasi *Smart Home* IoT.
- b. MultiChain sebagai aplikasi yang dapat memberikan akses ke *server private blockchain*.

3.4. Perancangan Topologi

3.4.1 Topologi Jaringan

Pada penelitian ini menggunakan topologi jaringan, yaitu terdapat *access point* untuk mejadi jaringan local 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 akan mengirimkan data kepada PC yang berperan sebagai *miner*. Topologi jaringan yang dibuat dapat dilihat pada Gambar 3.2 berikut ini :

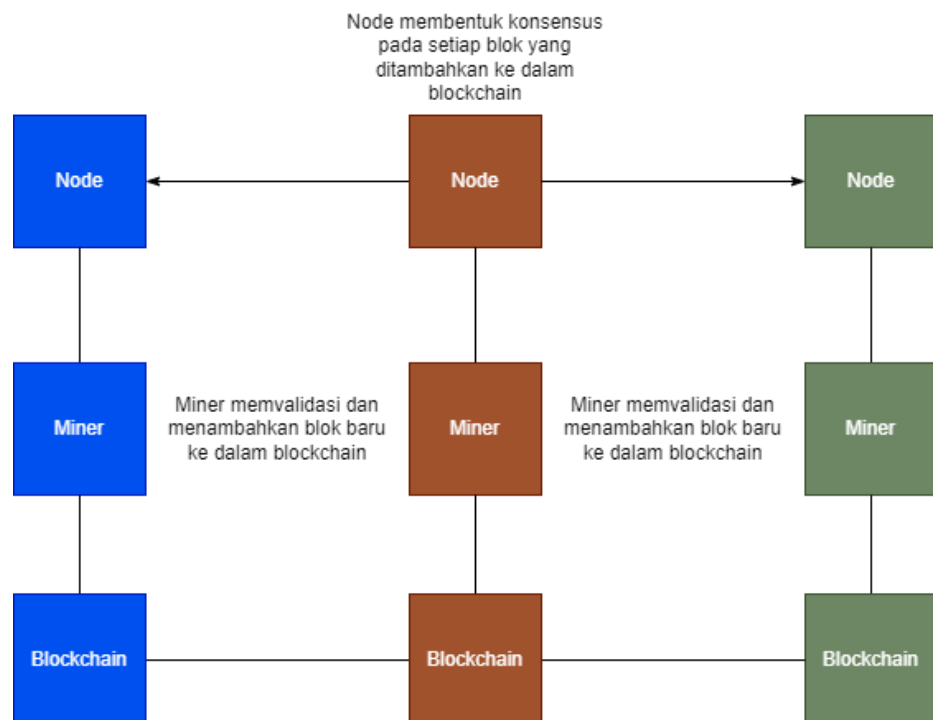


Gambar 3.2 Topologi Jaringan

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.

3.4.2 Alur Komunikasi *Blockchain*

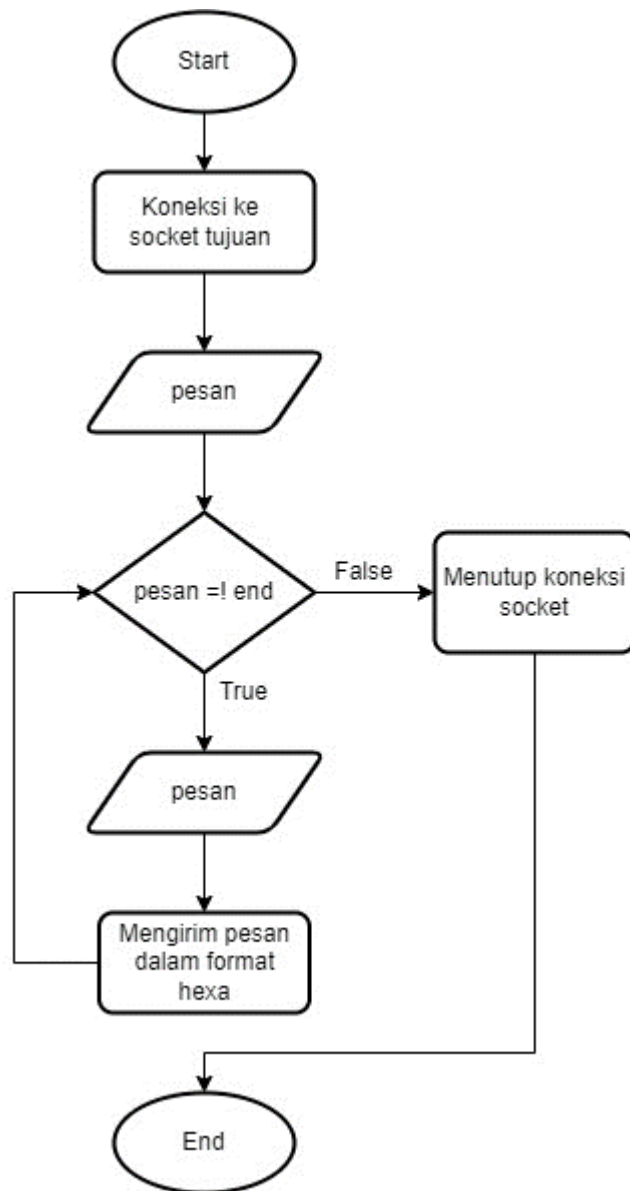
Peran *node* dan *miner* dalam jaringan *blockchain* yaitu *node* dapat saling berkomunikasi dengan sesama *node* dengan masing-masing *node* terdapat *miner* untuk memvalidasi transaksi dan menambahkan ke blok baru ke dalam *blockchain* seperti yang terlihat pada gambar dibawah ini :



Gambar 3.3 Alur Komunikasi *Blockchain*

3.4.3 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 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.



Gambar 3.4 Diagram Alir Program Simulasi Smart Home

3.4.4 Node admin

Berdasarkan Gambar.3.2 terdapat tiga *node* yang dimana akan dipilih satu *node* yang akan berperan sebagai *node admin*. *Node admin* akan berperan untuk menginisiasi *blockchain*, seperti memberi izin *node* yang diperbolehkan untuk terhubung ke dalam *blockchain*, memberi izin *node* untuk melakukan *mining* ke

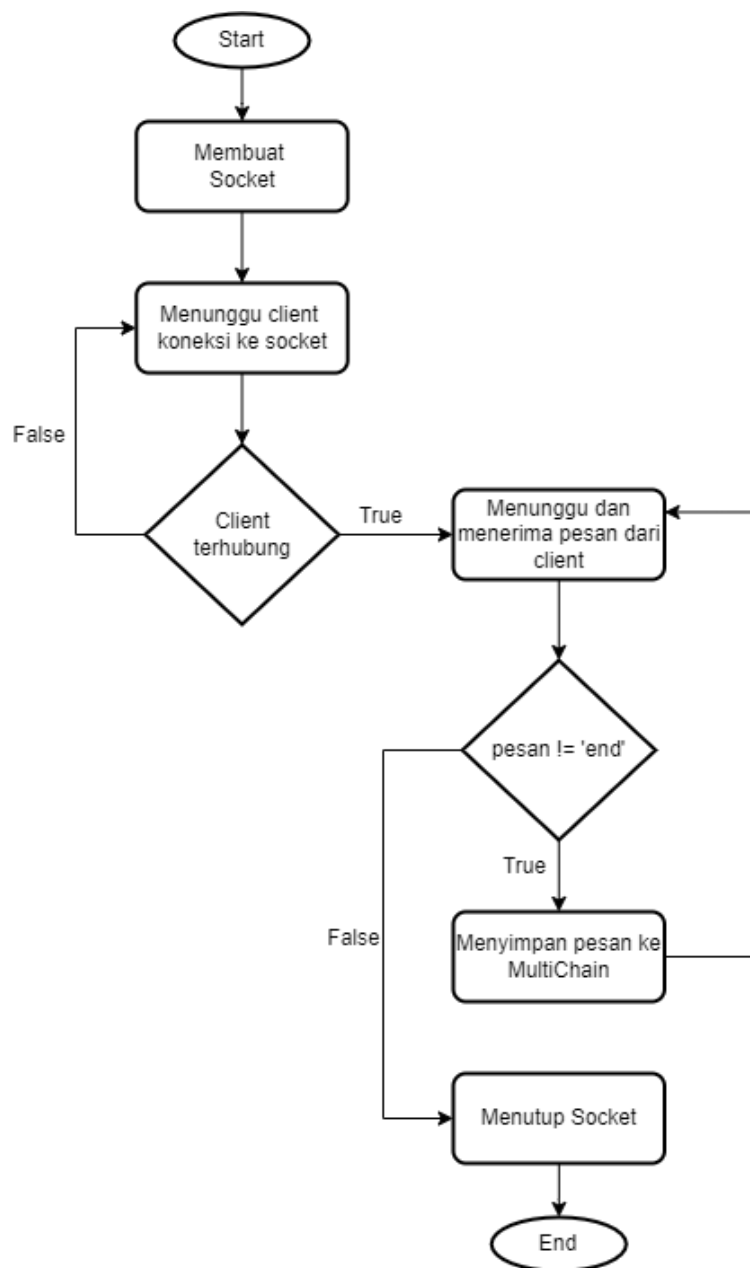
dalam *blockchain*, menjalankan sebuah program yang berfungsi sebagai penghubung antara *blockchain* pada MultiChain dengan perangkat *Smart Home* berbasis program yang akan mengirimkan paket data kemudian meneruskannya ke dalam *blockchain*. Untuk dapat menerima paket yang dikirim dari perangkat *smart home* perlu dibuat jalur komunikasi dengan menggunakan metode *socket programming* yang berfungsi sebagai penghubung antara *node admin* dengan perangkat *smart home* berbasis program *python* tersebut.

3.4.5 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* yang mengirimkan data kepada *node admin*. Setelah terhubung, perangkat *smart home* dapat mengirim data ke *node* MultiChain menggunakan *socket*. Data akan diserialisasi menjadi format yang sesuai *.JSON* atau format khusus, lalu kirim data melalui koneksi *socket* yang sudah dibuat. Dengan menggunakan perintah `send()` untuk mengirim data melalui *socket*. Data yang dikirimkan akan memiliki waktu jeda tiap pengiriman data adalah 25 detik karena harus menunggu proses *mining* dari *blockchain* terlebih dahulu.

3.4.6 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.



Gambar 3.5 Alur Penerimaan Data

3.5. Pengujian

3.5.1 Koneksi dengan MultiChain

Setelah pesan atau data diterima oleh *node admin*, kemudian data akan diteruskannya ke dalam *blockchain* dengan menggunakan API yang telah disediakan oleh MultiChain sehingga pesan tersebut dapat disimpan ke dalam *blockchain*. Untuk dapat berkomunikasi dengan API milik MultiChain dengan menggunakan bahasa pemrograman *python*, maka dapat memanfaatkan *library Savoir*.

3.5.2 Penyimpanan Data

Setelah terhubung dengan API milik MultiChain, maka untuk selanjutnya data dapat diterima dan disimpan ke dalam *blockchain*. Proses penyimpanan data disebut dengan proses *create*. 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*.

3.5.3 Validasi Data

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

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Multichain

Implementasi multichain dilakukan pada setiap node dengan melakukan proses install. Untuk proses create atau pembuatan jaringan *private blockchain* dapat dilakukan dengan perintah yang dijalankan pada terminal dengan CLI (Command Line Interface).

4.1.1. Proses Create Blockchain

Proses *create* atau pembuatan jaringan *private blockchain* dapat dilakukan dengan menjalankan perintah dengan CLI, dengan menjalankan perintah `multichain-util create blockchain_name`.



```
D:\TA\TAAAA\Multichain>multichain-util create iotchain

MultiChain 2.3.3 Utilities (latest protocol 20013)

Blockchain parameter set was successfully generated.
You can edit it in C:\Users\HP\AppData\Roaming\MultiChain\iotchain\params.dat before running multichaind for the first time.

To generate blockchain please run "multichaind iotchain -daemon".
```

Gambar 4.1 *Create Private blockchain*

Setelah perintah seperti gambar 4.1 maka *blockchain* akan membentuk parameter dan secara otomatis terbentuk folder sesuai dengan nama *blockchain* pada `C:\Users\HP\AppData\Roaming\Multichain\iotchain`. Sebelum melakukan *run blockchain* terlebih dahulu dapat dilakukan edit beberapa parameter pada lokasi folder *blockchain* berada. Secara *default* apabila *node* yang membentuk jaringan *private blockchain* akan menjadi *node admin*, yang semua inisiasi dan aksi dikontrol penuh oleh *node admin*.

4.1.2. Proses Run Blockchain

Proses *run blockchain* dapat dilakukan dengan perintah `multichaind blockchain_name -daemon` dimana pada saat pembuatan jaringan *private blockchain* pertama kali akan secara otomatis juga terbentuk struktur *blockchain*.

```
D:\TA\TAAAA\Multichain>multichaind iotchain -daemon

MultiChain 2.3.3 Daemon (Community Edition, latest protocol 20013)

Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind iotchain@192.168.9.20:7197

This host has multiple IP addresses, so from some networks:
multichaind iotchain@10.50.76.41:7197

Listening for API requests on port 7196 (local only - see rpcallowip setting)
Node ready.
```

Gambar 4.2 *Running Blockchain*

Seperti pada gambar 4.2 yang menunjukkan *blockchain* dijalankan kemudian terbentuk *genesis block*. *Genesis block* merupakan *block* pertama dari *blockchain*. Terdapat perintah `multichaind iotchain@192.168.9.20:7197` yang merupakan cara agar *node* lain dapat terhubung sebagai anggota dalam jaringan *private blockchain* yang telah dibuat.

4.1.3. Proses Penelusuran *Blockchain*

Proses penelusuran informasi *blockchain* dapat digunakan perintah yang diawali dengan `multichain-cli blockchain_name [option]`.


```

D:\TA\TAAAA\Multichain>multichain-cli iotchain getinfo
{"method":"getinfo","params":[],"id":"54798120-1699425576","chain_name":"iotchain"}
{
  "version" : "2.3.3",
  "nodeversion" : 20303901,
  "edition" : "Community",
  "protocolversion" : 20013,
  "chainname" : "iotchain",
  "description" : "MultiChain iotchain",
  "protocol" : "multichain",
  "port" : 7197,
  "setupblocks" : 60,
  "nodeaddress" : "iotchain@192.168.9.20:7197",
  "burnaddress" : "1XXXXXXbiXXXXXY1qXXXXXXWhXXXXXXa1FamN",
  "incomingpaused" : false,
  "miningpaused" : false,
  "offchainpaused" : false,
  "walletversion" : 60000,
  "balance" : 0,
  "walletdbversion" : 3,
  "reindex" : false,
  "blocks" : 3,
  "chainrewards" : 0,
  "streams" : 1,
  "timeoffset" : 0,
  "connections" : 0,
  "proxy" : "",
  "difficulty" : 5.96046447753906e-8,
  "testnet" : false,
  "keypoololdest" : 1699425559,
  "keypoolsize" : 2,
  "paytxfee" : 0,
  "relayfee" : 0,
  "errors" : ""
}

```

Gambar 4.3 Mendapatkan Info *Blockchain*

Pada gambar 4.3 didapatkan detail info dari *blockchain* dengan menjalankan perintah `multichain-cli blockchain_name getinfo`. Kemudian didalam *blockchain* terbentuk *blocks* sebanyak 3, yang dimana jumlah *block* akan terus bertambah apabila terjadi penambahan data pada jaringan *blockchain*.

```

D:\TA\TAAAA\Multichain>multichain-cli iotchain getblock 3
{"method":"getblock","params":["3"],"id":"45533616-1699425587","chain_name":"iotchain"}

{
  "hash" : "00d9d6be90de0317a4a0b625e42667cef43a20c05d5ea5b7db32e857ba981fbf",
  "miner" : "12zFm6fYXtuE4gQ1edakPbnSJw8YpTTahq7DVf",
  "confirmations" : 2,
  "size" : 265,
  "height" : 3,
  "version" : 3,
  "merkleroot" : "535849bac5528a505957543d939e02a6b17dd64e62763676b6327b6db1284554",
  "tx" : [
    "535849bac5528a505957543d939e02a6b17dd64e62763676b6327b6db1284554"
  ],
  "time" : 1699425569,
  "nonce" : 144,
  "bits" : "2000ffff",
  "difficulty" : 5.96046447753906e-8,
  "chainwork" : "0000000000000000000000000000000000000000000000000000000000000000",
  "previousblockhash" : "008623ec5e2ebf8839fdd944682f806e369526619b21e6275ae4635878964342",
  "nextblockhash" : "0062a0ee361b3a1fd5ff0d06172a3fc7b5a790d6693a0db46cbd14009fa24312"
}

```

Gambar 4.4 Mendapatkan Detail *Block*

Pada gambar 4.4 dijalankan perintah untuk mendapatkan detail dari struktur block yang berada pada *blockchain* dengan `multichain-cli blockchain_name getblock 3`. Dimana “*hash*” adalah nilai enkripsi dari *block* yang secara *runtime* akan terus berubah apabila *block* terus bertambah, “*miner*” adalah alamat yang terbentuk secara acak yang khusus digunakan untuk *node*.

4.1.4. Proses Mencari *Permissions*

```

D:\TA\TAAAA\Multichain>multichain-cli iotchain listpermissions
{"method":"listpermissions","params":[],"id":"75038300-1699425756","chain_name":"iotchain"}

[
  {
    "address" : "12zFm6fYXtuE4gQ1edakPbnSJw8YpTTahq7DVf",
    "for" : null,
    "type" : "mine",
    "startblock" : 0,
    "endblock" : 4294967295
  },
  {
    "address" : "12zFm6fYXtuE4gQ1edakPbnSJw8YpTTahq7DVf",
    "for" : null,
    "type" : "admin",
    "startblock" : 0,
    "endblock" : 4294967295
  },
]

```

Gambar 4.5 Mencari *List Permission* dalam *Blockchain*

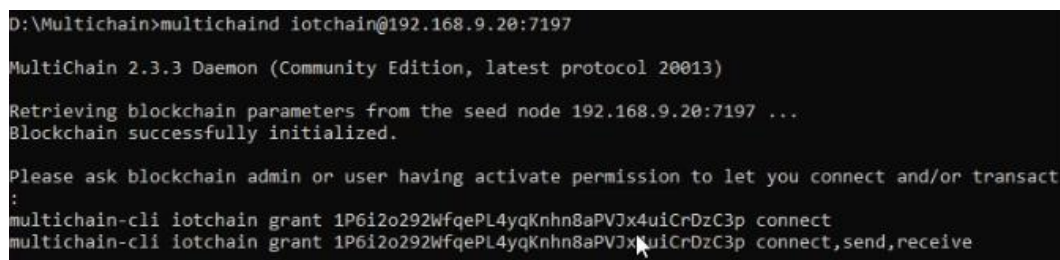
Perintah *list permissions* untuk mengetahui *address* dan *type* agar dapat mengetahui peran dari *address*, yang dilihat berdasarkan nilai *type* dengan

menjalankan perintah `multichain-cli blockchain_name listpermissions`. Kemudian didapatkan *address* dari *node* memiliki *type* sebagai *mine* dan *admin*.

4.2. Koneksi *Node client* dengan *Node admin*

4.2.1. Proses Koneksi oleh *Node Client*

Perintah untuk dapat terhubung dengan dengan *node* lainnya dapat menggunakan `multichaind blockchain_name@<seed-node-ip>:<seed-node-port>`.



```
D:\Multichain>multichaind iotchain@192.168.9.20:7197
MultiChain 2.3.3 Daemon (Community Edition, latest protocol 20013)
Retrieving blockchain parameters from the seed node 192.168.9.20:7197 ...
Blockchain successfully initialized.
Please ask blockchain admin or user having activate permission to let you connect and/or transact :
multichain-cli iotchain grant 1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p connect
multichain-cli iotchain grant 1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p connect,send,receive
```

Gambar 4. 6 *Node client* Mencoba Terhubung dengan Jaringan *Private blockchain*

Node client yang mencoba terhubung dengan jaringan *private blockchain* harus mendapatkan *permissions* oleh *Node admin* terlebih dahulu. Seperti yang terlihat pada gambar 4.6, dimana terdapat aksi yang dapat dilakukan *Node client* dapat berupa *connect*, *send*, *receive*.

4.2.2. Proses *Permissions* oleh *Node admin*

Perintah pemberian *permission* untuk terhubung yang berada pada *Node client* dapat diulangi untuk dijalankan pada *node admin* dengan `multichain-cli grant address_node-client connect,send,receive`.



```
D:\TA\TAAAA\Multichain>multichain-cli iotchain grant 1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p connect,send,receive
{"method": "grant", "params": [{"1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p", "connect,send,receive"}], "id": "88645283-1699425837", "chain_name": "iotchain"}
0326e92ea630ef5c5628ab699c89251be338a101fc1c9cb57558d56eea703e68
```

Gambar 4.7 Pemberian Izin untuk Terhubung oleh *Node admin*

Pada gambar 4.7 perintah *grant* diikuti dengan 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*

Pada gambar 4.9 dijalankan perintah `multichain-cli blockchain_name getblock 25` dan didapatkan jumlah *block* yang bertambah dan memberikan juga “*hash*” dari *block* itu sendiri dan nilai “*miner*” adalah alamat dari *node admin*.



```
C:\Windows\System32\cmd.exe
D:\TA\TAAAA\Multichain>multichain-cli iotchain getpeerinfo
{"method":"getpeerinfo","params":[],"id":"65199743-1699425962","chain_name":"iotchain"}
[
  {
    "id" : 3,
    "addr" : "192.168.9.21:50597",
    "addrlocal" : "192.168.9.20:7197",
    "services" : "0000000000000001",
    "lastsend" : 1699425956,
    "lastrecv" : 1699425956,
    "bytessent" : 12645,
    "bytesrecv" : 4322,
    "conntime" : 1699425895,
    "pingtime" : 0.099635,
    "version" : 70002,
    "subver" : "/MultiChain:0.2.0.13/",
    "handshakelocal" : "12zFm6fYXtuE4gQledakPbnSJw8YpTTahq70Vf",
    "handshake" : "1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p",
    "inbound" : true,
    "encrypted" : false,
    "startingheight" : 0,
    "banscore" : 0,
    "synced_headers" : -1,
    "synced_blocks" : -1,
    "inflight" : [
    ],
    "whitelisted" : false
  },
  {
    "id" : 4,
    "addr" : "10.50.76.32:7197",
    "addrlocal" : "10.50.76.41:50846",
    "services" : "0000000000000001",
    "lastsend" : 1699425960,
    "lastrecv" : 1699425960,
    "bytessent" : 2470,
    "bytesrecv" : 3837,
    "conntime" : 1699425899,
    "pingtime" : 0.106394,
    "version" : 70002,
    "subver" : "/MultiChain:0.2.0.13/",
    "handshakelocal" : "12zFm6fYXtuE4gQledakPbnSJw8YpTTahq70Vf",
    "handshake" : "1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p",
    "inbound" : false,
    "encrypted" : false,
    "startingheight" : 14,
    "banscore" : 0,
    "synced_headers" : 24,
    "synced_blocks" : -1,
    "inflight" : [
    ],
  },
]
```

Gambar 4.10 *Peer info* pada *Node admin*

```

    "id" : 2,
    "addr" : "192.168.9.20:7197",
    "addrlocal" : "192.168.9.21:50597",
    "services" : "0000000000000001",
    "lastsend" : 1699776880,
    "lastrecv" : 1699776880,
    "bytessent" : 9336,
    "bytesrecv" : 16417,
    "conntime" : 1699776722,
    "pingtime" : 0.109377,
    "version" : 70002,
    "subver" : "/MultiChain:0.2.0.13/",
    "handshakelocal" : "1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p",
    "handshake" : "12zFm6fYXtuE4gQ1edakPbnSJwBYpTTahq7DVf",
    "inbound" : false,
    "encrypted" : false,
    "startingheight" : 24,
    "banscore" : 0,
    "synced_headers" : 35,
    "synced_blocks" : 35,
    "inflight" : [
    ],
    "whitelisted" : false
  },
  {
    "id" : 3,
    "addr" : "10.50.76.41:50846",
    "addrlocal" : "10.50.76.32:7197",
    "services" : "0000000000000001",
    "lastsend" : 1699776880,
    "lastrecv" : 1699776880,
    "bytessent" : 8424,
    "bytesrecv" : 4215,
    "conntime" : 1699776726,
    "pingtime" : 0.089531,
    "version" : 70002,
    "subver" : "/MultiChain:0.2.0.13/",
    "handshakelocal" : "1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p",
    "handshake" : "12zFm6fYXtuE4gQ1edakPbnSJwBYpTTahq7DVf",
    "inbound" : true,
    "encrypted" : false,
    "startingheight" : 24,
    "banscore" : 0,
    "synced_headers" : 35,
    "synced_blocks" : 35,
    "inflight" : [
    ],
    "whitelisted" : false
  }
}

```

Gambar 4.11 *Peer Info pada Node Client*

Menggunakan perintah `multichain-cli blockchain_name getpeerinfo` dapat mengetahui jumlah *node* yang terhubung dengan jaringan *private blockchain* yang telah dibuat dan mendapatkan detail informasi dari setiap *node*. Dimana pada gambar 4.10 untuk *node admin* dan 4.11 untuk *node client*, kedua *node* dapat terhubung dengan menggunakan mekanisme *handshake*. Pada gambar nilai *handshake local* adalah alamat dari *node* dan nilai dari *handshake* adalah alamat *node* yang terhubung.

4.3. Membuat *Stream*

Stream adalah tempat menyimpan semua transaksi yang dilakukan di dalam jaringan *private blockchain* dan perintah untuk menyimpan transaksi data pada multichain dengan perintah *publish*.

4.3.1. *Create Stream*

Create streams dapat dilakukan dengan mengatur *mode restrict* dengan *write* dimana dapat dilakukan proses penyimpanan data secara langsung dengan perintah yang dijalankan `multichain-cli blockchain_name create stream stream_name "{\"restrict\":\"write\"}"`. Kemudian ditampilkan juga nilai *“txid”* dari berhasilnya proses *create stream* tersebut seperti pada gambar 4.12.

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain create stream iotdata "{\"restrict\":\"write\"}"
{"method":"create","params":["stream","iotdata",{ "restrict":"write"}],"id":"77886593-1699426355",
"chain_name":"iotchain"}
144943017341f550a9c28b1de47a1efa988084f66493119a3d0e7ed31c114b63
```

Gambar 4.12 *Create Stream*

4.3.2. *Mencari List Stream*

Untuk mencari *stream* yang telah di *create* dapat menggunakan perintah `multichain-cli blockchain_name liststreams`.

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain liststreams
{"method":"liststreams","params":[],"id":"31684926-1699426371","chain_name":"iotchain"}
{
  "name" : "iotdata",
  "createtxid" : "144943017341f550a9c28b1de47a1efa988084f66493119a3d0e7ed31c114b63",
  "streamref" : "55-265-18708",
  "restrict" : {
    "write" : true,
    "read" : false,
    "onchain" : false,
    "offchain" : false
  },
  "details" : {
  },
  "subscribed" : false
}
```

Gambar 4.13 *Mencari List Stream*

Pada gambar 4.13 didapatkan *stream* yang telah di *create* dengan nama *“iotdata”* dengan parameter *“createtxid”* sesuai dengan yang didapatkan pada saat membuat *stream*.

4.3.3. Menyimpan Data pada Stream

Menyimpan data ke dalam *stream* dapat dilakukan dengan menggunakan perintah *publish*, dengan dapat menjalankan perintah `multichain-cli blockchain _name publish stream_name key_value data_value`.

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain publish iotdata key1 "{\"json\":{\"Suhu\":\"31C\", \"Kelembaban\":\"50%\"}}\"
{\"method\":\"publish\",\"params\":[\"iotdata\",\"key1\",{\"json\":{\"Suhu\":\"31C\", \"Kelembaban\":\"50%\"}}],\"id\":\"28938261-1699426520\", \"chain_name\":\"iotchain\"}
169c290460489328337f0020532ab1d734fb96411b5236dc2ce81c79f72a5f6e
```

Gambar 4.14 *Publish Data pada Stream*

Pada gambar 4.14 dilakukan penyimpanan data ke dalam *stream* dengan mode *write*, yang dimana dilakukan *input* data secara langsung dengan CLI. Kemudian didapatkan nilai “txid” dari proses *publish* data tersebut.

4.3.4. Melihat Detail Data pada Stream

Perintah untuk menampilkan detail data pada *stream* dapat menggunakan perintah `multichain-cli blockchain_name liststreamitems stream_name`.

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain liststreamitems iotdata
{"method":"liststreamitems","params":["iotdata"],"id":"91150242-1699426553", "chain_name":"iotchain"}
error code: -703
error message:
Not subscribed to this stream
```

Gambar 4.15 Gagal Melihat Detail *Items*

Pada gambar 4.15 *Node* tidak dapat melihat data pada *stream* karena belum melakukan proses *subscribe* pada *stream* yang telah dibuat.

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain subscribe iotdata
{"method":"subscribe","params":["iotdata"],"id":"19558397-1699426570", "chain_name":"iotchain"}
```

Gambar 4.16 Proses *Subscribe Stream*

Perintah *subscribe* dapat dijalankan dengan `multichain-cli blockchain_name subscribe stream_name`.


```

D:\TA\TAAAA\Multichain>multichain-cli iotchain liststreamitems iotdata true 1
{"method":"liststreamitems","params":["iotdata",true,1],"id":"79485152-1699426596","chain_name":"iotchain"}

[
  {
    "publishers" : [
      "12zFm6fYXtuE4gQ1edakPbnSJwBYpTTahq7DVf"
    ],
    "keys" : [
      "key1"
    ],
    "offchain" : false,
    "available" : true,
    "data" : {
      "json" : {
        "Suhu" : "31C",
        "Kelembaban" : "50%"
      }
    },
    "confirmations" : 5,
    "blockhash" : "003188b21133118207d4c7e5cb589e62582e570a9f0fd229c56f9ef5c8c71c08",
    "blockheight" : 66,
    "blockindex" : 1,
    "blocktime" : 1699426532,
    "txid" : "169c290460489328337f0020532ab1d734fb96411b5236dc2ce81c79f72a5f6e",
    "vout" : 0,
    "valid" : true,
    "time" : 1699426520,
    "timereceived" : 1699426520
  }
]

```

Gambar 4.17 Detail *Items* dalam *Stream*

Kemudian perintah `multichain-cli blockchain_name liststreamitems stream_name` dapat dijalankan kembali, maka didapatkan detail item yang disimpan didalam stream beserta nilai “txid” sesuai dengan yang didapatkan pada saat proses *publish* data seperti pada gambar 4.14.

4.4. Penyimpanan Data *Node Client* pada *Stream*

4.4.1. *Node client* Mencari Detail Item pada *Stream*

```

D:\Multichain>multichain-cli iotchain liststreamitems iotdata
{"method":"liststreamitems","params":["iotdata"],"id":"97335734-1699777473","chain_name":"iotchain"}

error code: -703
error message:
Not subscribed to this stream

```

Gambar 4.18 *Node client* Gagal Mendapat Detail *Items*

Pada gambar 4.18 *Node client* tidak dapat melihat detail *item* pada *stream* karena belum melakukan proses *subscribe* pada *stream*.

4.4.2. Node client Gagal Menyimpan Data pada Stream

```
D:\Multichain>multichain-cli iotchain publish iotdata key2 "{\"json\":{\"Suhu\":\"32C\", \"Kelembaban\":\"40%\"}}\"
{\"method\":\"publish\",\"params\":[\"iotdata\",\"key2\",{\"json\":{\"Suhu\":\"32C\", \"Kelembaban\":\"40%\"}}],\"id\":\"
67026276-1699777577\",\"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 4.19 Node client Gagal Menyimpan Data

Pada gambar 4.19 *Node client* tidak dapat menyimpan data pada *stream* karena belum memiliki *permission* untuk melakukan *publish* data ke dalam *stream*.

4.4.3. Permission oleh Node admin

```
D:\TA\TAAAA\Multichain>multichain-cli iotchain grant 1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p send
{"method":"grant","params":["1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p","send"],\"id\":\"32591326-16994
26846\",\"chain_name\":\"iotchain\"}

5a58e5b54a7da451d7758a5fd02645fcf69aa79d6fa07aa68f1c2a48007e31f4

D:\TA\TAAAA\Multichain>multichain-cli iotchain grant 1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p iotda
ta.write
{"method":"grant","params":["1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p","iotdata.write"],\"id\":\"23326
822-1699426857\",\"chain_name\":\"iotchain\"}

89241019d0c24fa912feed7c62c8482ba296f261fd06cb386ad2ad5054cb383e
```

Gambar 4.20 Pemberian Izin oleh Node admin

Perintah untuk memberikan *permissions* oleh *node admin* dengan menjalankan perintah `multichain-cli blockchain_name grant node_client_address send`, dimana diberi *permissions* 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* seperti terlihat pada gambar 4.20.

4.4.4. Node Client Berhasil Menyimpan Data pada Stream

```
D:\Multichain>multichain-cli iotchain publish iotdata key2 "{\"json\":{\"Suhu\":\"32C\", \"Kelembaban\":\"40%\"}}\"
{\"method\":\"publish\",\"params\":[\"iotdata\",\"key2\",{\"json\":{\"Suhu\":\"32C\", \"Kelembaban\":\"40%\"}}],\"id\":\"
85093844-1699777711\",\"chain_name\":\"iotchain\"}

5255299887453932aa9351cd7cbe23bf159fe6b69da21490a25c93cd6e1d84d8
```

Gambar 4.21 Node client Berhasil Menyimpan Data

Pada gambar 4.21 *Node client* berhasil menyimpan data pada *stream* setelah diberi *permissions* oleh *node admin*. Kemudian proses penyimpanan data disertai dengan *txid* diberikan setelah melakukan *publish* data ke dalam *stream*.

4.4.5. Pencarian Detail Item pada *Stream*

```
D:\Multichain>multichain-cli iotchain subscribe iotdata
{"method":"subscribe","params":["iotdata"],"id":"31684926-1699777823","chain_name":"iotchain"}

D:\Multichain>multichain-cli iotchain liststreamitems iotdata
{"method":"liststreamitems","params":["iotdata"],"id":"59360332-1699777829","chain_name":"iotchain"}

[
  {
    "publishers": [
      "12zFm6fYXtuE4gQ1edakPbnSJw8YpTTahq7DVf"
    ],
    "keys": [
      "key1"
    ],
    "offchain": false,
    "available": true,
    "data": {
      "json": {
        "Suhu": "31C",
        "Kelembaban": "50%"
      }
    },
    "confirmations": 23,
    "blocktime": 1699426532,
    "txid": "169c290460489328337f0020532ab1d734fb96411b5236dc2ce81c79f72a5f6e"
  },
  {
    "publishers": [
      "1P6i2o292WfqePL4yqKnhn8aPVJx4uiCrDzC3p"
    ],
    "keys": [
      "key2"
    ],
    "offchain": false,
    "available": true,
    "data": {
      "json": {
        "Suhu": "32C",
        "Kelembaban": "40%"
      }
    },
    "confirmations": 8,
    "blocktime": 1699426894,
    "txid": "5255299887453932aa9351cd7cbe23bf159fe6b69da21490a25c93cd6e1d84d8"
  }
]
```

Gambar 4.22 Proses *Subscribe* dan Mendapatkan Detail *Items*

Pada gambar 4.22 dilakukan terlebih dahulu proses *subscribe* dengan perintah `multichain-cli blockchain_name subscribe iotdata` oleh *Node client* agar dapat melihat detail item pada *stream*. Kemudian dilakukan poses mencari detail items dengan perintah `multichain-cli blockchain_name liststreamitems iotdata` dan didapatkan dengan *key1* adalah data dari hasil *publish* oleh *node admin* dan *key2* adalah hasil *publish* oleh *node client*.

4.5. Program Perangkat *Smart Home*

Program simulasi *smart home* dijalankan dari perangkat komputer, *output* dari simulasi program *smart home* kemudian akan disimpan kedalam jaringan *private blockchain*.

```
import socket
import random
import time
import binascii

server_ip = "192.168.9.20"
server_port = 4000

addr = (server_ip, server_port)

def send_data(data):
    try:
        client_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
        client_socket.connect(addr)
        size = len(data.encode())

        client_socket.send(data.encode())
        print("Bytes size :", size, "Bytes")
        print("Hex data:", data)

        response = client_socket.recv(1024).decode("utf-8")
        print(response)

        client_socket.close()
    except Exception as e:
        print("Gagal mengirim data ke server:", str(e))

if __name__ == "__main__":
    lights_on = False
    temperature = 20.0
    humidity = 50.0
    motion_detected = False
    for hour in range(24):
        temperature += random.uniform(-1.0, 1.0)
        humidity += random.uniform(-5.0, 5.0)
        motion_detected = random.choice([True, False])

        if hour >= 6 and hour < 8:
            lights_on = True
        elif hour >= 18 and hour < 22:
            lights_on = True
        else:
            lights_on = False

        if motion_detected:
            temperature += 1.0
            humidity += 10.0
    try:
```

```

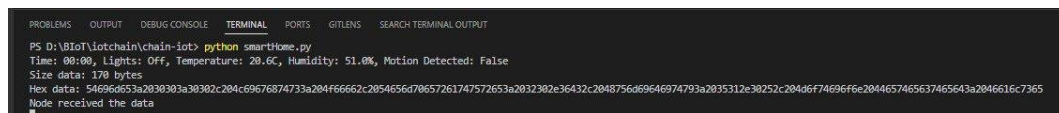
        data = "Time: {:02d}:00, Lights: {}, Temperature:
{:01.1f}C, Humidity: {:01.1f}%, Motion Detected: {}" .format(
            hour, "On" if lights_on else "Off", temperature,
            humidity, motion_detected)
        hex_data = binascii.hexlify(data.encode()).decode()
        print(data)
        send_data(hex_data)

        time.sleep(3600)

    except ConnectionAbortedError as e:
        print("Error koneksi:", e)
        break

```

Pada *script* diatas adalah program simulasi yang bertindak sebagai klien. Simulasi program akan terhubung dengan program *server* karena digunakan *socket programming* yang memberi nilai alamat IP dan *Port* dari program *server*. Proses simulasi *smart home* diatur dengan data akan dikirim ke program *server* dengan `time.sleep(3600)` atau dikirim setiap jam.



Gambar 4.23 Hasil Simulasi Program *Smart Home*

Pada gambar 4.23 adalah *output* dari simulasi program *smart home* yang akan dikirim ke program *server* dimana nilai asli dari simulasi diubah terlebih dahulu kedalam format *hexadecimal* agar dapat disimpan oleh program *server* ke dalam multichain.

4.6. Program *Server* Untuk Terhubung dengan Multichain

```

import socket
from Savoir import Savoir

rpcuser = 'multichainrpc'
rpcpassword = 'hq5xxpMVQB771DZtUhmFZMmkbhrDJv9y9eqHCWXzELB'
rpchost = '192.168.9.20'
rpcport = '7196'
chain_name = 'iotchain'

multichain = Savoir(rpcuser, rpcpassword, rpchost, rpcport,
chain_name)

```

```

server_ip = "192.168.9.20"
server_port = 4000

addr = (server_ip, server_port)

server_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
server_socket.bind(addr)
server_socket.listen(5)
print(f"Server mendengarkan di {addr}")

def handle_recv_data():
    response = 'Node received the data'
    client_socket.send(response.encode())

    client_socket.close()

while True:
    client_socket, client_address = server_socket.accept()
    print("Menerima koneksi dari", client_address)

    data = client_socket.recv(1024).decode()
    print("Menerima data:", data)

    handle_recv_data()

    try:
        txid = mc.publish("iotdata", "key2", data)
        print("Data telah disimpan dengan TXID:", txid)

    except Exception as e:
        print(f"Error: {e}")

```

Pada *script* diatas merupakan program *server* yang akan menerima koneksi dari klien untuk mendapat data hasil simulasi hasil program *smart home* kemudian akan dilakukan proses *publish* data ke dalam *stream*.

```

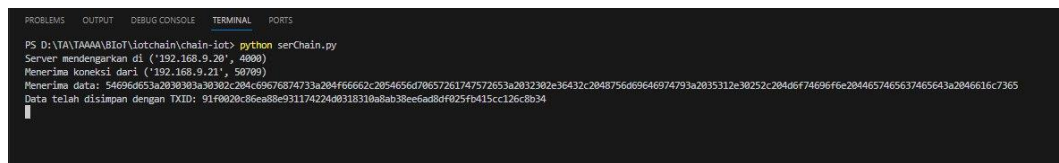
from Savoir import Savoir

rpcuser = 'multichainrpc'
rpcpassword = 'hq5xxpMVQB771DZtUhmFZMmkbhrDJv9y9eqHCWXzELB'
rpchost = '192.168.9.20'
rpcport = '7196'
chain_name = 'iotchain'

multichain = Savoir(rpcuser, rpcpassword, rpchost, rpcport,
chain_name)

```

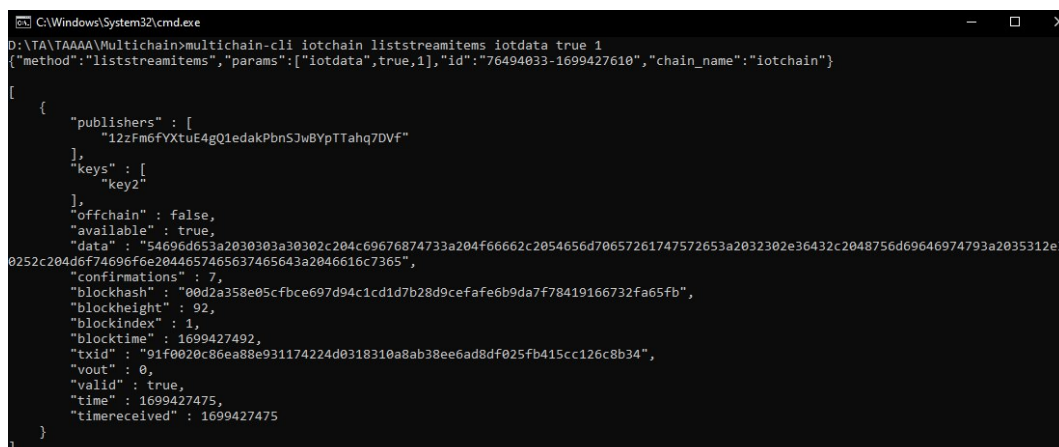
Pada bagian kode diatas adalah konfigurasi program *server* dengan multichain dengan menggunakan API JSON RPC. Konfigurasi program *server* dengan multichain menggunakan *library Savoir* dimana untuk mengambil nilai dari API JSON RPC dari multichain agar dapat terhubung dengan multichain.



Gambar 4. 24 Menerima dan *Publish* Data oleh Server

Pada gambar 4.24 adalah hasil dari program *server* 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 dilakukan proses *publish* data ke dalam *stream* pada multichain.

4.7. Pencarian Data pada Stream



Gambar 4.25 Details *Items* dari Data *Smart Home*

Pada gambar 4.25 adalah pencarian data 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 4.24 dan data diberikan juga keterangan *timestamp*.

4.8. *Permissionless Device*

Permissionless device adalah perangkat yang tidak dapat melakukan aksi yang sama seperti *permissioned device* dan beberapa perilaku *permissionless device* digambarkan seperti:

4.8.1. *Permissionless Device* Mencoba Terhubung

```
C:\Windows\System32\cmd.exe
D:\Titip\multichain-windows-2.3.3>multichaind iotchain@10.50.76.80:7197
MultiChain 2.3.3 Daemon (Community Edition, latest protocol 20013)
Retrieving blockchain parameters from the seed node 10.50.76.80:7197 ...
Blockchain successfully initialized.
Please ask blockchain admin or user having activate permission to let you connect and/or transact:
multichain-cli iotchain grant 1CHWScumNuXLmk3VgR7xV3sFznkE2nh83UwVcw connect
multichain-cli iotchain grant 1CHWScumNuXLmk3VgR7xV3sFznkE2nh83UwVcw connect,send,receive
```

Gambar 4.26 *Permissionless Device* Mencoba Terhubung

Pada gambar 4.26 *permissionless device* mencoba terhubung dengan jaringan blockchain dan didapatkan pemberitahuan untuk meminta izin kepada node admin pada saat menjalankan perintah `multichaind blockchain_name@<seed_ip_node>:<seed_port_node> .`

4.8.2. *Permissionless Device* Menyimpan Data

```
D:\Titip\multichain-windows-2.3.3>multichain-cli iotchain publish iotdata key1 "{\"json\":{\"Suhu\":\"31C\",\"Kelembaban\":\"49%\"}}\"
error: Could not connect to the server 127.0.0.1:7196 (error code 1 - "EOF reached")
Make sure the multichaind server is running and that you are connecting to the correct RPC port.
```

Gambar 4.27 *Permissionless Device* Gagal Menyimpan Data

Pada gambar 4.27 *permissionless device* mencoba menyimpan 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* tidak dapat terhubung dengan *server* multichain tetapi melakukan penyimpanan data.

4.8.3. *Permissionless Device* Mencari Detail Items

```
D:\Titip\multichain-windows-2.3.3>multichain-cli iotchain listsreamitems true 1
error: Could not connect to the server 127.0.0.1:7196 (error code 1 - "EOF reached")
Make sure the multichaind server is running and that you are connecting to the correct RPC port.
```

Gambar 4.28 *Permissionless Device* Gagal Mencari Detail Item

Pada gambar 4.28 *permissionless device* mencoba mendapatkan detail item dari percobaan *permissionless device* saat menyimpan data pada gambar 4.27

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.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan diperoleh beberapa kesimpulan sebagai berikut:

1. *Blockchain* dapat menjadi salah satu cara untuk meningkatkan keamanan data IoT, dengan dilakukan integrasi *blockchain* dengan IoT data pada sistem IoT menjadi lebih terlindungi dari terjadinya manipulasi karena dengan didukungnya oleh *blockchain* yang memiliki struktur sistem yang terdesentralisasi dan terenkripsi.
2. *Blockchain* memberikan integritas data dapat dijaga dengan lebih baik, setiap dilakukan proses transaksi data IoT dapat diverifikasi dan diotorisasi dengan aman dengan adanya hash value yang diberikan setiap melakukan transaksi dan juga bukti *txid* pada setiap dilakukannya penyimpanan data.
3. Menggunakan Jaringan *private blockchain* dimana akses untuk tergabung ke dalam jaringan yang terkendali, yang berarti bahwa pihak-pihak tertentu saja yang memiliki otorisasi yang dapat mengakses dan terlibat dalam jaringan tersebut. Menjadikan tingkat keamanan dan privasi serta memastikan bahwa akses data hanya dibuka bagi node yang memiliki izin.

5.2. Saran

Berdasarkan penelitian ini, terdapat beberapa kekurangan dan potensi pengembangan yang dapat menjadi referensi bagi penelitian selanjutnya, diantaranya sebagai berikut:

1. Pada penelitian selanjutnya diharapkan dapat menggunakan platform atau aplikasi *blockchain* yang lebih ampuh dalam menjaga integritas data dan meningkat keamanan data pada lingkungan IoT, serta mempelajari lebih banyak tentang penggunaan efektif dari jaringan *private blockchain*.

2. Pada penelitian selanjutnya diharapkan dalam mendistribusikan data IoT dapat menggunakan mekanisme yang lebih baik dari penelitian ini dan mempelajari lebih dalam tentang *networking* yang terjadi pada saat mendistribusikan data tersebut.

DAFTAR PUSTAKA

- Alam S., R., Jain S., and Doriya R. (2021). *Security threats and solutions to IoT using Blockchain: A review.Proccedings – 5th Internasional Conference on Intelligent Computing and Control System, ICICCS 2021.*
- Andersson, Karl et al. *LCN Symposium 2019 : 2019 IEEE 44th Local Computer Networks Symposium on Emerging Topics in Networking : Proceedings : 14 October 2019, Osnabrück, Germany.*
- Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P. (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.*
- Efendi, Yoyon. 2018. “Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile.” *Jurnal Ilmiah Ilmu Komputer* 4(2): 21–27.
- Hayes, A., (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>
- J. Frankenfield, 2019, “Proof of Work”: Investopedia, 28 june 2020. [online]. Available:<https://www.investopedia.com/terms/p/proofwork.asp#:~:text=P roof%20of%20work%20describes%20a,launching%20denial%20of%20service%20attacks>.
- J. Frankenfield, 2019, “ Proof of Stake” Invetopedia, 11 Agustus 2019. [online]. Available:<https://www.investopedia.com/terms/p/proof-of-stake-pos.asp>.
- Khan, M.A., Salah, K., (2018). IoT Security: Review, *blockchain* solution, and open challenges. Future generation computer system.

- Mingyu, H., Tianyu, K., & Li, G., (2020). *A Blockchain Based Architecture for IoT Data Sharing Systems*. Jurnal Beijing University of Posts and Telecommunications Key Lab of Universal Wireless Communications, Ministry of Education.
- Nesterenko R., V., dan Maslova M., A. (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.
- Pahlajani, Sunny, Avinash Kshirsagar, and Vinod Pachghare. "Algorithms." *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*: 1–6.
- Panarello A, Tapas N, Merlino G, Longo F, Puliafito A, 2018, *Blockchain* and IoT integration: A systematic survey. Sensors (Switzerland).Andersson, Karl et al. *LCN Symposium 2019: 2019 IEEE 44th Local Computer Networks Symposium on Emerging Topics in Networking: Proceedings: 14 October 2019, Osnabrück, Germany*.
- Premkumar, R., and Priya S. 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.
- Rumah. (2023). [online]. Apa itu MulthiChain di *Blockchain*. Artikel rumah. Available:https://id.tishijie.com/10609/Apa_itu_MultiChain_di_blockchan
- Seth, S., (2022) Explained Crypto in Cryptocurrency. [online]. Available: <https://www.investopedia.com/tech/explaining-crypto-cryptocurrency/>
- Udin, Ikram et al. 2019. "The Internet of Things: A Review of Enabled Technologies and Future Challenges." *IEEE Access* 7: 7606–40.