Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)
Vol.11, No.2, April 2024, hlm.235-242
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019

DOI: 10.25126/jtiik.20241126108
p-ISSN: 2355-7699
e-ISSN: 2528-6579

SIMULASI PENGGUNAAN BLOCKCHAIN PADA KEAMANAN JARINGAN *INTERNET OF THINGS* MENGGUNAKAN PIN *EMULATOR*: MODEL *PUBLIC* BLOCKCHAIN

Clara Amalia Ismayanti[1], Parma Hadi Rantelinggi*[2]

[1,2]Universitas Papua, Manokwari
Email: [1]claraamel53@gmail.com, [2]p.rantelinggi@unipa.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 07 Februari 2022, diterima untuk diterbitkan: 25 April 2024)

Abstrak

Internet of Things (IoT) adalah sejumlah perangkat yang dapat mengumpulkan dan mengirimkan data antar sensor tanpa perlu bantuan manusia. Namun, keamanan IoT dapat terancam karena sifatnya yang dapat diakses dari mana saja dan kapan saja. Celah keamanan memiliki kemungkinan sukar untuk dideteksi, karena memiliki pola yang beragam. Oleh karena itu, diperlukan sebuah model keamanan pengiriman data antar sensor yang aman. Blockchain adalah teknologi yang dapat menjawab tiga syarat keamanan yang diperlukan, yaitu ketersediaan, kerahasiaan, dan integritas. Dalam penelitian ini, kami mencoba membangun sebuah simulasi keamanan IoT dengan menggunakan teknologi blockchain, dimana sistem keamanannya menggunakan pola *public* blockchain. Metode yang digunakan dalam penelitian ini adalah model *public* blockchain yang memungkinkan para pengguna untuk mengendalikan aplikasi yang terhubung pada pin *emulator* melalui *smart contract*. Penelitian ini menggunakan jejaring Ethereum yang termasuk dalam jaringan pengujian yang dapat digunakan tanpa adanya biaya transaksi yang perlu dibayar. Dengan adanya penelitian ini, diharapkan dapat memberikan solusi pada permasalahan keamanan dan tantangan yang dihadapi dalam jaringan IoT.

Kata kunci: *blockchain, internet of things (IoT), simulasi,keamanan, smart contract*

BLOCKCHAIN SYSTEM SIMULATION MODEL ON INTERNET OF THINGS NETWORK SECURITY

Abstract

The Internet of Things (IoT) is a collection of devices that can collect and transmit data between sensors without the need for human assistance. However, IoT security can be threatened due to its accessibility from anywhere and at any time. Security vulnerabilities may be difficult to detect due to their diverse patterns. Therefore, a secure data transmission security model between sensors is needed. Blockchain is a technology that can meet the three security requirements needed, namely availability, confidentiality, and integrity. In this study, we attempt to build a security simulation of IoT using blockchain technology, where the security system uses a public blockchain pattern. The method used in this study is a public blockchain model that allows users to control applications connected to the pin emulator via smart contracts. This research uses the Ethereum network, which is included in the testing network that can be used without transaction fees. With this research, it is hoped that solutions can be provided for the security issues and challenges faced in IoT networks.

Keywords: *blockchain, internet of things (IoT), simulation, security, smart contract*

1. PENDAHULUAN

Internet of Things (IoT) adalah sejumlah perangkat yang mampu mengumpulkan dan mengirimkan data antar sensor tanpa perlu bantuan manusia (Ali et al., 2019). Namun, sifat IoT yang dapat diakses dari mana saja dan kapan saja, maka hal ini dapat menimbulkan ancaman pada keamanannya.

Celah keamanan memiliki kemungkinan sukar untuk dideteksi, karena memiliki pola yang beragam. Penyebab adanya ancaman pada keamanan bisa disebabkan oleh konfigurasi jaringan yang kurang tepat (Sidiq et al., 2020; Riadi et al., 2021). Prinsip dari aspek sistem keamanan jaringan secara umum terdiri dari tiga hal yaitu ketesediaan, kerahasiaan, dan integritas (Matondang et al., 2018; Pramudita et al., 2020; Sari et al., 2020). Maka diperlukan sebuah

model keamanan pengiriman data antar sensor yang aman.

Blockchain adalah teknologi yang mampu menjawab tiga syarat keamanan yang di perlukan yaitu ketersediaan, kerahasiaan dan integritas. Blockchain adalah teknologi baru yang awal penerapannya pada *bitcoin* (Nakamoto, 2008), yang terus berkembang dengan cepat dalam satu dekade ini (Huang et al., 2021). Blockchain pada dasarnya adalah sistem basis data terdistribusi yang menyimpan data transaksional yang diamankan oleh kriptografi (Taylor et al., 2020). Selain itu, blockchain menggunakan kriptografi saat memproses serta memverifikasi sebuah proses transaksi. (Cole et al., 2019). Enkripsi dan pengkodean data dalam blockchain meningkatkan transparansi, efisiensi dan kepercayaan dalam berbagi data dan informasi.

Dibeberapa bidang, teknologi blockchain telah mempengaruhi dan memberikan manfaat di bidang pertanian (Wihartiko et al., 2021), kesehatan, farmasi (Fernando et al., 2020), ekonomi, industry (Helliar et al., 2020) dan bidang lainnya. Terkhusus dalam bidang teknologi, yaitu jaringan untuk IoT dan sensor. Dimana manajemen dan pengaturan IoT menggunakan private blockchain (Košťál et al., 2019; Lockl et al., 2020). Sistem keamanan mengusulkan sebuah private blockchain-*based access control* yang melibatkan penggunaan *private* blockchain dalam menyediakan keamanan dasar yang tidak bisa di manipulasi (Xue et al., 2018). Sehingga memberikan tantangan tersendiri dalam menghadirkan keamanan dan solusi pada permasalahan jaringan IoT (Singh et al., 2021).

Dalam penelitian ini, kami mencoba untuk membangun sebuah simulasi keamanan IoT, dimana sistem keamanannya menggunakan teknologi blockchain dengan mengambil pola *public* blockchain. Akses aplikasi dibangun menggunakan *text editor* Visual Studio Code dan dikendalikan dengan menggunakan pin *emulator* GPIO. Pin *emulator* GPIO memiliki fungsi untuk membaca status pin, mengatur/menghapus pin, dan menjalankan panggilan balik setelah interupsi pin (Feng et al., 2020). Dalam penelitian kami, kami mengamati setiap pin dalam melakukan transaksi data. Python sudah menyediakan *package* pin *emulator* GPIO yang siap digunakan dengan cara menginstal menggunakan pip. Oleh sebab itu, pada penelitian ini kami menggunakan pin *emulator* GPIO karena sifatnya yang sudah tersedia untuk digunakan. Selain itu, *emulator* ini banyak digunakan dalam penelitian lainnya dibandingkan dengan *emulator* lain yang masih jarang digunakan. Untuk pola integrasi blockhain dan IoT memanfaatkan pola integrasi *asset* ke blockchain. Program yang dibangun kemudian dijalankan pada *private network* kemudian pengguna dapat melihat status setiap pin pada baris perintah dan juga pada *emulator*.

Penelitian seperti ini sudah pernah dilakukan sebelumnya: penelitian yang dilakukan oleh Puri et al. (2021) menghasilkan pendekatan yang diusulkan untuk menunjukkan kinerja yang memuaskan dan performa yang berhubungan dengan *bandwidth* dan konsumsi daya. Dalam penelitian kami memiliki kesamaan dengan penelitian yang dilakukan oleh Puri et al. dimana kami ingin melihat kinerja konsumsi daya. Tetapi pada penelitian kami juga akan melihat transaksi dan *block time* pada saat transaksi dilakukan. Selain itu, penelitian Baccelli et al. (2018) menghasilkan ketersediaan sistem operasi yang *open source* untuk perangkat *Low-end Embedded* dalam IoT seperti RIOT. Dalam penelitian ini menggunakan GPIO sebagai *micro controller*. Pada penelitian kami, kami menggunakan pin *emulator* GPIO tetapi kami juga mengintegrasikan GPIO dengan *controller* berbasis web agar pengguna dapat menggunakan *controller* yang lebih mudah dipahami untuk memberikan interupsi terhadap pin pada *emulator*.

Penggunaan blockchain pada simulasi keamanan IoT melalui pin *emulator* adalah untuk mengevaluasi kelayakan keamanan sebelum penerapan yang sebenarnya dilakukan. Selain itu, kami menggunakan *simulator* agar dapat mensimulasikan *leader*, *server*, dan perangkat untuk *generate* dan mengirim transaksi. Dengan menggunakan blockchain maka perangkat dapat mengumpulkan dan bertukar data dengan server, perangkat lain, maupun *platform* lainnya. Keuntungan dari blockchain untuk meningkatkan sistem IoT adalah: desentralisasi, verifikasi kolektif dan ketahanan terhadap gangguan, keamanan pribadi, kecepatan, penghematan biaya, dan *smart contract* (Zhou et al., 2018; Atlam et al., 2018).

Pada penelitian ini, kami mengadopsi model *public* blockchain, dimana setiap pengguna yang memiliki akses internet dan akses aplikasinya dapat melakukan transaksi pada setiap aplikasi yang terhubung dengan mengendalikan pin *emulator* GPIO *(turning on* atau *turning off*) melalui *smart contract*. Dengan begitu, para pengguna dapat mengontrol aplikasi yang terhubung pada pin *emulator*. Selain itu, penelitian ini menggunakan jejaring Ethereum yang termasuk dengan jaringan pengujian yang dapat kami gunakan tanpa adanya biaya transaksi yang perlu dibayar (Sidiq et al., 2020). Dasar penggunaan model *public* blockchain pada penelitian ini karena bersifat *anonymous*. Meskipun data yang digunakan tersimpan pada jaringan *puclic* blockchain bersifat transparan atau dapat dilihat oleh pihak lain. Namun, untuk identitas setiap pengguna yang mengirim atau menerima data transakasi tidak dapat diketahui oleh pihak lainnya (Fadhillah et al., 2022).

2. METODE PENELITIAN

2.1. Model Blockchain

Penggunaan blockchain untuk sebuah keamanan mengacu pada satu set penambang terdesentralisasi yang menjalankan protokol konsensus yang aman. Konsensus pada blockchain yang didistribusikan dan tidak dapat dipercaya, maka dapat dianggap *Byzantine Generals Problem* (BGP). Jika hal ini terjadi, maka pembuat program kemudian bertanggung jawab untuk memvalidasi kembali blok baru dan mendistribusikan melalui jaringan yang telah ditetapkan. Dalam penelitian ini, kami menggunakan mekanisme konsensus *Proof of Stake* (PoS). PoS digunakan untuk bukti pertaruhan *asset crypto* dalam Ethereum. Pada tabel 1 memperlihatkan perbandingan protokol konsensus yang ada pada blockchain (Panarello et al., 2018).

Selain mempertahankan buku besar (*public ledger*) global untuk keseimbangan disetiap nama samaran, blockchain juga dapat mengeksekusi program yang telah dibuat dan ditentukan oleh pengguna. Seperti waktu, negara bagian publik, pengiriman pesan, nama samaran, kebenaran serta ketersediaannya (Kosba et al., 2016).

2.2. Pola Integrasi Blockchain dan IoT

Integrasi antara blockchain dan IoT terdapat empat pola yang dapat digunakan. Penggunaannya tergantung pada kebutuhan pengguna masing-masing. Pada gambar 1 merupakan gambaran dari pola integrasi dan IoT, terdiri atas empat pola integrasi yang umum digunakan. Bentuk integrasi selalu dimulai dari asset yang dikategorikan dalam *field*, kemudian diintegrasikan dengan jaringan melalui IoT *cloud* atau *gateway*/fog. Kemudian diteruskan ke tahapan berikut yaitu *backend*.

Dari keempat pola tersebut, penelitian kami menggunakan pola integrasi nomor 4, yaitu *Asset* → Blockchain. Dalam hal ini, *asset* pada program aplikasi yang telah dibuat akan langsung dihubungkan dengan blockchain.

2.3. Perancangan Program

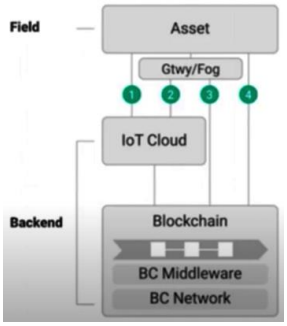
Pada program yang dibuat dijalankan dengan menggunakan GPIO *emulator*, terminal (*Command Prompt* pada Windows) dan juga *controller* untuk *pin* yang berbasis website untuk antar-muka. Desain antar muka dapat dilihat pada gambar 2.

Melalui perancangan yang telah dikerjakan, maka dapat diketahui diagram alir skematik dari penelitian ini (gambar 3). Pada diagram alir penelitian ini dapat diketahui bahwa ketika proses telah dimulai, hal yang dilakukukan terlebih dahulu ialah menjalankan *ganache-cli*.

Kemudian, jalankan *truffle compile* (Windows), lalu jalankan aplikasi Python. Jika tidak berhasil, maka coba jalankan kembali aplikasi Python. Jika aplikasi Python berhasil dijalankan, maka pengguna dapat melanjutkan dengan membuka *controller* dan jalankan pin (*on/off*). Jika pin tidak berhasil dijalankan, maka buka kembali *controller* atau lakukan *refresh* pada *controller*. Jika pin berhasil dijalankan, maka pada masing-masing terminal akan menampilkan riwayat transaksi yang berlangsung dan proses selesai. Kebutuhan perangkat dan bahasa pemograman pada penelitian ini dapat dilihat pada tabel 2.

Tabel 1. Perbandingan Protokol Konsensus Pada Blockchain

Kategori	PoW	PoS	PoET	BFT and Variants	Federated BFT
(i) Blockchain type	Permissionless	Both	Both	Permissionless	Permissionless
(ii) Transaction finality	Probabilistic	Probabilistic	Probabilistic	Immediate	Immediate
(iii) Transaction rate	Low	High	Medium	High	High
(iv) Token needed?	Yes	Yes	No	No	No
(v) Cost of participation	Yes	Yes	No	No	No
(vi) Scability of peer network	High	High	High	Low	High
(vii) Trust model	Untrusted	Untrusted	Untrusted	Semi-trusted	Semi-trusted



Gambar 1. Pola Integrasi Blockchain dan IoT

Tabel 2. Kebutuhan perangkat dan bahasa pemgraman

No	Parangkat	Kategori
1	Windows build tool	Compiler *source code*
2	GPIO emulator	Emulator untuk membaca I/O
3	Flask	Web *Framework*
4	Web3	Media aplikasi web
5	Python	Bahasa Pemograman
6	HTML	Membuat struktur halaman
7	CSS	Mengatur tampilan
8	Javacript	Bahasa pemograman

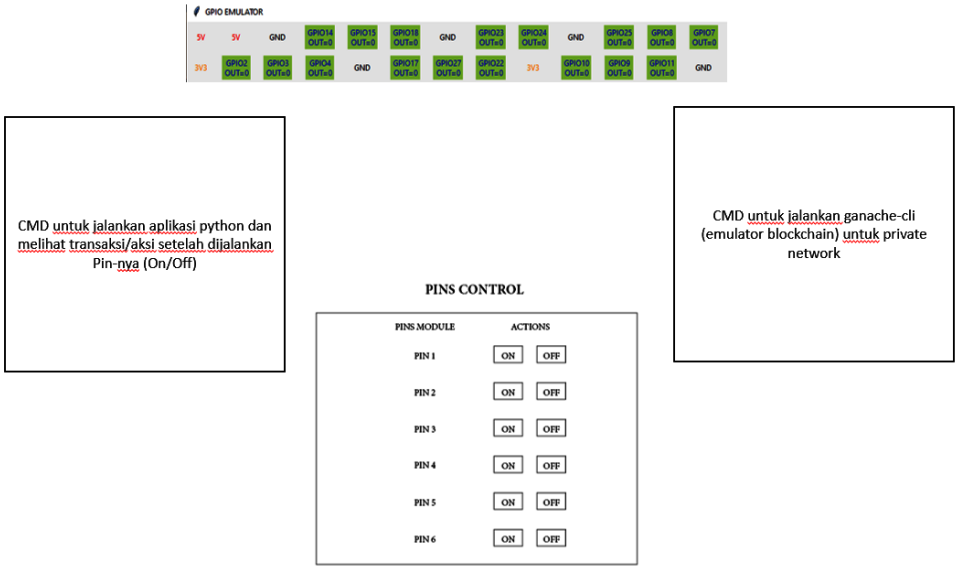
3. HASIL DAN DISKUSI

Untuk membuat program ini, kami menggunakan *text editor* Visual Studio Code serta menggunakan *source code* yang terdapat pada akun Github Salman Dabbakuti (Dabbakuti, 2021), tetapi kami juga melakukan modifikasi pada *controller* dengan menggunakan rancangan kami. Program yang telah dibuat, kemudian dijalankan pada *private network* dan pengguna dapat melihat status setiap pin pada baris perintah dan juga pada *emulator*. Dimana pada dasarnya hasil akan menampilkan status pin tertentu pada blockchain dan kemudian mengaktifkan konfigurasi pin yang sesuai.

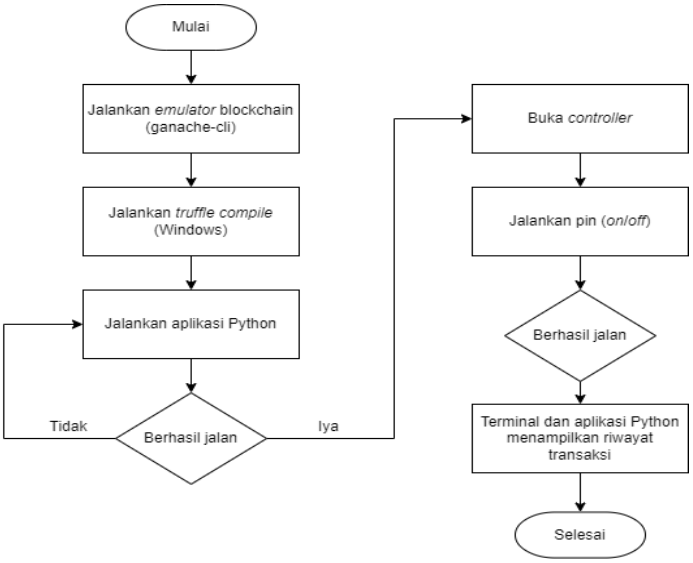
3.1. File Konfigurasi

Metadata konfigurasi mendefinisikan sebuah alur kerja tingkat tinggi dan model interaksi pada aplikasi blockchain. Untuk konfigurasi yang dilakukan terdapat pada file JSON (PatAltimore, 2022). Terdapat dua jenis file, yaitu *home automation* dan *migrations* pada penelitian kami.

Peran konfigurasi yang dilakukan adalah untuk mendefinisikan tindakan yang terjadi dalam blockchain. Dalam hal ini akan terjadi skenario permintaan - tanggapan.



Gambar 2. Desain Antar Muka



Gambar 3. Diagram Alir Skematik

3.2. File Code Smart Contract

Pada *smart contract*, Ethereum menggunakan *Solidity* sebagai bahasa pemrograman untuk menuliskan logika. *Smart contract* dalam file *Solidity* hampir sama dengan *object-oriented*. Pada program kami, terdapat dua file kontrak, yaitu *Migrations* dan *Pin Control*.

a. Versi Pragma

Versi pragma yang kami gunakan ialah pragma solidity ^0.5.0. Untuk versi ini sudah cukup baik dan sudah cocok digunakan pada penelitian kami. Solidity merupakan bahasa pemrograman berorientasi obyek yang digunakan untuk menuliskan *smart contract* dan mengimplementasikannya.

b. Code Migrations.sol

Truffle mengharuskan pengguna memiliki kontrak migrasi. Pada folder *contracts* terdapat file migrations.sol. Kemudian pada folder mogrations terdapat file 1_initial_migration.js yang mengirimkan kontrak migrations.sol ke Ethereum blockchain. Berikut merupakan *source code* dari file migrations.sol:

```
contract Migrations {
  address public owner;
  uint256 public last_completed_migration;

  constructor() public {
  owner = msg.sender;
   }

  modifier restricted() {
  if (msg.sender == owner) _;
  }

   function setCompleted(uint256 completed) public restricted
{
   last_completed_migration = completed;
  }

   function upgrade(address new_address) public restricted {
   Migrations upgraded = Migrations (new_address);
   upgraded.setCompleted (last_completed_mig ration);
   }
 }
```

c. Code pinControl.sol

Untuk *contract* selanjutnya terdapat file pinControl.sol yang berguna untuk kontrak pin. Berikut merupakan *source code* dari file pinControl.sol

3.3. File Aplikasi Python

Untuk aplikasinya, kami menggunakan bahasa pemrograman Python karena telah menyediakan berbagai *library* yang diperlukan. Pada demo program menggunakan program yang tidak menggunakan *key* dan kami juga menyediakan satu file aplikasi yang mengharuskan menggunakan *private keys*.

3.4. Pin Controller

Dalam mengontrol setiap pin pada emulator, kami membuat *controller* berbasisis website dengan *Front-End* yang dibangun dengan bahasa HTML, CSS dan Javascript. Dalam membuat *controller* ini juga dibantu dengan menggunakan *framework* Bootstrap. Antar muka dari *controller* yang telah dibuat dapat dilihat pada gambar 4.

```
contract homeAutomation {
    address owner;

    constructor() public {
      owner = msg.sender;
    }

    struct pin {
      uint256 status;
    }

    mapping(uint256 => pin) public pinStatus;

    function control(uint256 _pin, uint256 _status)  public {
      require(msg.sender == owner);
      pinStatus[_pin].status = _status;
    }

  }
```

3.5. Hasil Demonstrasi Program

a. *Instantiate* ganache-cli *private network*

Pada awal demonstrasi, hal pertama yang dilakukan oleh kami yaitu menjalankan ganache-cli *private network* pada terminal (dalam hal ini kami menggunakan *command prompt* pada sistem operasi Windows). Saat dijalankan maka akan menampilkan 10 akun yang tersedia dan 10 private keys yang dapat digunakan. Selain itu, pada terminal juga tertampil *gas price*, *gas limit*, *call gas limit*, dan juga itu berjalan pada port:8545.

b. *Compile contract*

Setelah menjalankan ganache-cli, selanjutnya jalankan *truffle compile* (untuk sistem operasi Windows) pada *command prompt*. Hal ini dilakukan untuk mengkompilasi *contract* yang ada pada direktori *contracts* dan membangun artefak pada direktori *build*.

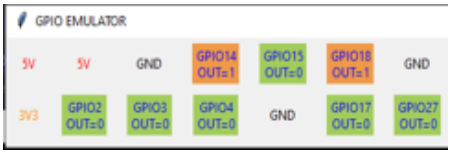
c. Jalankan aplikasi Python

Kemudian, jalankan aplikasi Python yang tidak menggunakan *private key* untuk melakukan pengujian (*public)* pada *private network*. Setelah melakukan *deployment contracts* pada *private network*, maka kita dapat melihat status setiap pin dibaris perintah dan pada *emulator*. Setelah aplikasi Python sudah dijalankan, maka transaksi akan mulai terhitung.

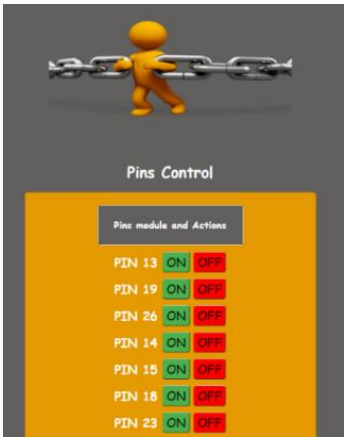
d. Nyalakan atau matikan pin

Pengguna dapat melakukan *turn on* atau *off* pin melalui *controller*. Pada simulasi ini, kami mencoba melakukan *turn on* pada pin 14 dan pin 18.

Setelah melakukan *turn on* pada pin 14 dan pin 18 melalui *controller*, maka status pin pada GPIO emulator berubah, yang dapat dilihat pada gambar 3. Pada emulator terlihat perubahan warna dan status pin berubah menjadi 1.

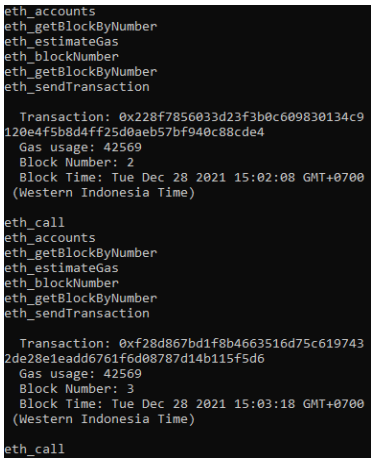


Gambar 3. Status Pin Pada *Emulator* Mengalami Perubahan



Gambar 4. Antar Muka Pin *Controller* Berbasis Website

Kemudian, pengguna juga dapat melihat detail transaksi pin 14 dan pin 18 yang sudah dinyalakan. Gambar 5 merupakan detail transaksi yang ditampilkan melalui terminal. Untuk detail transaksi kita dapat melihatnya melalui terminal yang menjalankan ganache-cli, detail transaksi memperlihatkan *transaction*, *gas usage*, *block number*, dan *block time*. Seperti yang telah dijelaskan pada model blockchain sebelumnya, pada detail transaksi dapat melihat waktu, negara bagian publik, pengiriman pesan, nama samaran, kebenaran serta ketersediaannya. Tetapi tidak dapat melihat identitas asli pengguna dan data yang dikirim atau diterima tidak dapat diketahui oleh pihak lain karena data yang sudah dienkripsi menjadi sebuah kode dan hanya dapat diketahui oleh pengirim atau penerima saja.



Gambar 5. Detail transaksi saat pin 14 dan 18 dinyalakan

Pada terminal juga dapat terlihat status pin berubah menjadi 1. Pada status *pin*, logika yang terdapat pada aplikasi yaitu 0 dan 1, 0 artinya *pin* dalam keadaan mati dan 1 artinya pin dalam keadaan menyala.

3.6. Diskusi

Penelitian ini membutuhkan sebuah skema pengiriman data yang mudah dimengerti atau dipahami, serta memerlukan data yang dapat dilaporkan dengan nilai *hash* pada data dan tercatat pada jejaring blockchain. Pada bagian *controller*, perlu menambahkan fitur ketika *button turning on* atau *off* diklik, *button* tersebut dapat berubah warna, sehingga pada *controller* dapat terlihat lebih jelas pin yang dinyalakan atau dimatikan.

Pada penelitian ini memiliki metodologi yang belum cukup signifikan sehingga diperlukan penggabungan metodologi lainnya agar dapat menemukan kebaruan penelitian yang berbeda dengan penelitian lainnya. Selain itu, penelitian ini masih memiliki kekurangan untuk perhitungan dan analisis. Sehingga untuk penelitian selanjutnya, diharapkan dapat mengukur perbandingan maupun perhitungan dalam penggunaan blockchain di sebuah penelitian.

4. KESIMPULAN

Kesimpulan dari penelitian ini adalah implementasi blockchain pada perangkat IoT dapat meningkatkan keamanan data pada saat transaksi dilakukan karena sifatnya yang *anonymous*, tidak dapat melihat identitas asli pengguna dan data yang dikirim atau diterima tidak dapat diketahui oleh pihak lain karena data yang sudah dienkripsi menjadi sebuah kode dan hanya dapat diketahui oleh pengirim atau penerima saja. Sehingga data penting yang terdapat pada perangkat bisa terhindar dari berbagai ancaman yang dilakukan oleh pihak-pihak yang tidak diinginkan. Pada penelitian ini, implementasi blockchain berlangsung dengan menggunakan *controller* untuk mengontrol semua pin yang terdapat pada *emulator*. Sehingga pada saat pin dinyalakan atau dimatikan akan menampilkan riwayat transaksi pada setiap terminal.

DAFTAR PUSTAKA

ALI, G., AHMAD, N., CAO, Y., ASIF, M., CRUICKSHANK, H. & ALI, Q.E., 2019. Blockchain based permission delegation and access control in Internet of Things (BACI). *Computers & Security*, 86, pp.318–334. https://doi.org/10.1016/j.cose.2019.06.010.

ATLAM, H. F., ALENEZI, A., ALASSAFI, M. O., & WILLS, G. B. (2018). Blockchain with Internet of Things: Benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications*, *10*(6), 40–48. https://doi.org/10.5815/ijisa.2018.06.05

BACCELLI, E., GUNDOGAN, C., HAHM, O., KIETZMANN, P., LENDERS, M. S.,

PETERSEN, H., SCHLEISER, K., SCHMIDT, T. C., & WAHLISCH, M. (2018). RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. *IEEE Internet of Things Journal*, *5*(6), 4428–4440. https://doi.org/10.1109/JIOT.2018.2815038

COLE, R., STEVENSON, M. & AITKEN, J., 2019. Blockchain technology: implications for operations and supply chain management. *Supply Chain Management: An International Journal*, 24(4), pp.469–483. https://doi.org/10.1108/SCM-09-2018-0309.

DABBAKUTI, S., 2021. *Salmandabbakuti/IoT-and-Blockchain*. [Python] Available at: <https://github.com/Salmandabbakuti/IoT-and-Blockchain> [Accessed 25 Jan. 2022].

FADHILLAH, Y. et al. (2022). *TEKNOLOGI BLOCKCHAIN DAN IMPLEMENTASINYA*. Medan: Yayasan Kita Menulis

FENG, B., MERA, A., & LU, L. (2020). *P²IM: Scalable and Hardware-independent Firmware Testing via Automatic Peripheral Interface Modeling.* https://www.usenix.org/conference/usenixsecurity20/presentation/feng

FERNANDO, E., MEYLIANA, H. & WARNARS, E.A., 2020. Blockchain technology for pharmaceutical drug distribution in Indonesia: A proposed model. *ICIC Express Letters*, 14(2), pp.113–120.

HELLIAR, C.V., CRAWFORD, L., ROCCA, L., TEODORI, C. & VENEZIANI, M., 2020. Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, 54, p.102136. https://doi.org/10.1016/j.ijinfomgt.2020.102136.

HUANG, H., KONG, W., ZHOU, S., ZHENG, Z. & GUO, S., 2021. A Survey of State-of-the-Art on Blockchains: Theories, Modelings, and Tools. *ACM Computing Surveys*, 54(2), p.44:1-44:42. https://doi.org/10.1145/3441692.

KOSBA, A., MILLER, A., SHI, E., WEN, Z. & PAPAMANTHOU, C., 2016. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In: *2016 IEEE Symposium on Security and Privacy (SP)*. 2016 IEEE Symposium on Security and Privacy (SP). pp.839–858. https://doi.org/10.1109/SP.2016.55.

KOŠŤÁL, K., HELEBRANDT, P., BELLUŠ, M., RIES, M. & KOTULIAK, I., 2019. Management and Monitoring of IoT Devices Using Blockchain. *Sensors*, 19(4), p.856. https://doi.org/10.3390/s19040856.

LOCKL, J., SCHLATT, V., SCHWEIZER, A., URBACH, N. & HARTH, N., 2020. Toward Trust in Internet of Things Ecosystems: Design Principles for Blockchain-Based IoT Applications. *IEEE Transactions on Engineering Management*, 67(4), pp.1256–1270. https://doi.org/10.1109/TEM.2020.2978014.

MATONDANG, N., NURLAILI ISNAINIYAH, I., & MULIAWATI, A. (2018). *Analisis Manajemen Risiko Keamanan Data Sistem Informasi (Studi Kasus: RSUD XYZ). 2*(1), 282–287. http://jurnal.iaii.or.id

NAKAMOTO, S., 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, p.21260.

PANARELLO, A., TAPAS, N., MERLINO, G., LONGO, F. & PULIAFITO, A., 2018. Blockchain and IoT Integration: A Systematic Survey. *Sensors*, 18(8), p.2575. https://doi.org/10.3390/s18082575.

PATALTIMORE, 2022. *Create a blockchain application - Azure Blockchain Workbench - Azure Blockchain*. [online] Available at: <https://docs.microsoft.com/en-us/azure/blockchain/workbench/create-app> [Accessed 25 Jan. 2022].

PRAMUDITA, R., FUADA, S., & MAJID, N. W. A. (2020). Studi Pustaka Tentang Kerentanan Keamanan E-Learning dan Penanganannya. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *4*(2), 309. https://doi.org/10.30865/mib.v4i2.1934

PURI, V., PRIYADARSHINI, I., KUMAR, R., & VAN LE, C. (2021). Smart contract based policies for the Internet of Things. *Cluster Computing*, *24*(3), 1675–1694. https://doi.org/10.1007/s10586-020-03216-w

RIADI, I., UMAR, R. & LESTARI, T., 2021. Smart Payment Application Security Optimization from Cross-Site Scripting (XSS) Attacks Based on Blockchain Technology. *Telematika*, 14(2), pp.74–85. https://doi.org/10.35671/telematika.v14i2.1221.

SARI, I.Y. et al. (2020). *KEAMANAN DATA DAN INFORMASI.* Medan: Yayasan Kita Menulis

SIDIQ, M.F., BASUKI, A.I., FIRDAUS, H. & BAIHAQI, M.A., 2020. Sentralisasi Pengawasan Informasi Jaringan Menggunakan Blockchain Ethereum. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(6), pp.1187–1196. https://doi.org/10.25126/jtiik.2020722662.

SINGH, S., HOSEN, A.S.M.S. & YOON, B., 2021. Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *IEEE Access*, 9, pp.13938–13959. https://doi.org/10.1109/ACCESS.2021.3051602.

TAYLOR, P.J., DARGAHI, T., DEHGHANTANHA, A., PARIZI, R.M. & CHOO, K.-K.R., 2020. A systematic literature review of blockchain cyber security. *Digital Communications and Networks*, 6(2), pp.147–156. https://doi.org/10.1016/j.dcan.2019.01.005.

WIHARTIKO, F.D., NURDIATI, S., BUONO, A. & SANTOSA, E., 2021. Blockchain dan Kecerdasan Buatan dalam Pertanian : Studi Literatur. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(1), pp.177–188. https://doi.org/10.25126/jtiik.0814059.

XUE, J., XU, C. & ZHANG, Y., 2018. Private Blockchain-Based Secure Access Control for Smart Home Systems. *KSII Transactions on Internet and Information Systems (TIIS)*, 12(12), pp.6057–6078. https://doi.org/10.3837/tiis.2018.12.024.

ZHOU, L., WANG, L., SUN, Y., & LV, P. (2018). BeeKeeper: A Blockchain-Based IoT System with Secure Storage and Homomorphic Computation. *IEEE Access*, *6*, 43472–43488. https://doi.org/10.1109/ACCESS.2018.2847632