Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 5, Mei 2019, hlm. 5154-5164 http://j-ptiik.ub.ac.id

Pengembangan Sistem Penyimpanan Data Sensor IoT Berbasis *Permissioned Blockchain* dengan menggunakan *Platform* Hyperledger

Arya Wardhana Budi Utomo[1], Adhitya Bhawiyuga[2], Kasyful Amron[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]aryawardana1997@gmail.com, [2]bhawiyuga@ub.ac.id, [3]kasyful@ub.ac.id

Abstrak

Penggunaan teknologi IoT dalam kehidupan sehari-hari secara tidak langsung akan memproduksi banyak data yang mengandung unsur privasi data pengguna. Penggunaan *cloud* untuk menyimpan data IoT saat ini menimbulkan ancaman privasi data pengguna yang disebabkan oleh pengelolaan data yang diserahkan sepenuhnya kepada pihak *cloud provider*. Oleh sebab itu, dibutuhkan suatu sistem penyimpanan data yang dapat menjaga privasi data pengguna sehingga ancaman terhadap privasi data pengguna dapat diminimalisir. Salah satu teknologi yang dapat dimanfaatkan untuk menyimpan data dengan menjaga privasi data adalah *blockchain.* Sebagai salah satu solusi untuk menyelesaikan masalah ancaman privasi data penguna dalam pengelolaan data secara terpusat pada *cloud*, penelitian ini membuat sistem penyimpanan data sensor IoT yang menerapkan *blockchain* dengan jenis *permissioned blockchain* dengan menggunakan *platform* Hyperledger. Hyperledger dipilih untuk memperoleh waktu validasi penyimpanan yang cepat. Hasil dari implementasi penelitian ini adalah sistem dapat menyimpan, merubah, dan mengambil data sensor IoT pada *blockchain.* Kemudian, privasi data pengguna terjaga dengan kontrol akses dan *channel* yang terdapat dalam *blockchain.* Pengujian *latency*, menunjukkan waktu penyimpanan data semakin bertambah seiring dengan jumlah data sensor IoT yang akan disimpan.

Kata kunci: *internet of things, permisioned blockchain, privasi, hyperledger fabric, hyperledeger composer*.

Abstract

The use of IoT technology in everyday life will produce a lot of data containing user data privacy. In this era the use of cloud to store IoT data involves the privacy of user data related to data management provided by cloud providers. Therefore, a data storage system that can protect the privacy of user data is needed so that the threat of privacy of user data can be minimized. One technology that can be used to store data and protect data privacy is blockchain. As one solution to solve the privacy issues of user data in data management centered on the cloud, this study create an IoT sensor data storage system that applies permissioned blockchain by using the Hyperledger platform. Hyperledger choosed to get short storage validation time. The results of the implementation of this research are systems that can store, change, and take IoT data sensors on the blockchain. Then, user privacy data is maintained with access controls and channels on the blockchain. Latency test shows the data storage time is increase in accordance with the number of IoT data sensors that will be saved.

Keywords: *Internet of Things, Permisioned Blockchain, privacy, Hyperledger Fabric, Hyperledeger Composer.*

1. PENDAHULUAN

Peningkatan penggunaan teknologi IoT *(Internet of Things)* saat ini membuat kebutuhan akan tempat penyimpanan data IoT semakin bertambah. IoT merupakan sebuah jaringan yang mempunyai kemampuan untuk memantau dan mengontrol lingkungan fisik di berbagai tempat melalui internet dengan cara mengumpulkan dan memproses data yang dihasilkan oleh sensor atau *smart device* (Rahman, et al., 2016). Dengan kemampuan yang dimilikinya, IoT semakin banyak diterapkan dalam aktivitas kehidupan sehari-hari manusia seperti *Smart Home*, *E-Health*, *Military* dan *Surveillance*, dll. Beberapa penerapan IoT tersebut secara tidak langsung akan memproduksi banyak data yang mengandung unsur privasi data pengguna.

Namun, secara umum IoT merepresentasikan perangkat-perangkat kecil dengan kemampuan penyimpanan data yang terbatas. Oleh sebab itu, kebanyakan dari perangkat IoT memanfaatkan *cloud* untuk melakukan penyimpanan data IoT (Liu, et al., 2017).

Cloud dapat memberikan kapasitas penyimpanan yang besar untuk data IoT (Aazam, et al., 2014). Namun, penyimpanan data dengan jumlah besar yang dikelola secara terpusat oleh *single cloud provider* dapat menimbulkan banyak persoalan (Rifi, et al., 2017). Salah satunya adalah persoalan privasi data pengguna yang disebabkan oleh pengelolaan data yang diserahkan sepenuhnya kepada pihak *cloud provider* (Wei, et al., 2013). Hal tersebut dapat menjadi ancaman terhadap privasi data pengguna karena pengguna tidak mengetahui bagaimana pengelolaan dan pendistribusian data oleh pihak *cloud*. Privasi merupakan hak pengguna untuk menjaga kerahasiaan dan kontrol atas informasi yang dimiliknya ketika diberikan kepada pihak lain (Porambage, et al., 2016). Oleh karena itu, dibutuhkan suatu sistem penyimpanan data yang dapat memberikan pengguna informasi dalam pengelolaan data sehingga ancaman terhadap privasi data pengguna dapat diminimalisir.

Salah satu teknologi yang dapat dimanfaatkan untuk menyimpan data dengan menjaga privasi data adalah *blockchain. Blockchain* merupakan sebuah struktur data terdistribusi yang direplikasi dan disebarkan diantara seluruh anggota dalam *blockchain* (Christidi & Devetsikiotis, 2016). Konsensus menjadikan *blockchain* tidak memiliki otoritas terpusat dan kontrol pengelolaan data menjadi terdistribusi di setiap anggota *blockchain* (Wang, et al., 2016). Setiap penambahan atau perubahan data ke dalam *blockhain* akan tercatat dan diketahui oleh seluruh anggota *blockchain.* Sehingga seluruh anggota *blockchain* memiliki salinan data yang sama dan meningkatkan rasa kepercayaan dalam *blockchain* (Makhdoom, et al., 2018). Pencatatan data dan pendistribusian data dalam *blockchain* tersebut dapat bermanfaat untuk menjaga privasi data pengguna. Selain itu, pengaturan kontrol akses pada *blockchain* juga dapat diterapkan dalam menjaga privasi data.

Pemanfaatan teknologi *blockchain* dalam lingkup IoT telah dilakukan oleh Ayoade dengan mengimplementasikan *platform* Ethereum *blockchain* (Ayoade, et al., 2018). Ayoade telah berhasil menerapkan *blockchain* untuk mencatat dan mengatur hak akses data IoT. Namun, proses validasi atau kesepakatan pencatatan data yang diterapkan oleh *platform* Ethereum membutuhkan waktu yang cukup lama dengan menggunakan konsensus *Proof of Work* (PoW). Oleh karena itu, berdasarkan hasil penelitian yang telah dilakukannya, Ayoade menyarankan untuk menggunakan *blockchain* dengan jenis *permissioned* yang memiliki algoritma validasi pencatatan data dengan waktu yang lebih cepat dibandingkan dengan konsensus PoW. *Permissioned blockchain* merupakan suatu jenis *blockchain* yang membatasi jumlah anggota atau *node* dengan mengetahui identitas dari node tersebut (Makhdoom, et al., 2018).

Sebagai salah satu solusi untuk menyelesaikan masalah ancaman privasi data penguna dalam pengelolaan data secara terpusat pada *cloud*, penelitian ini akan membuat sistem penyimpanan data sensor IoT yang menerapkan *blockchain* dengan menggunakan *platform* Hyperledger. Hyperledger merupakan salah satu *platform blockchain* yang berjenis *permissioned* (Androulaki, et al., 2018). Hyperledger dapat menerapkan mekanisme validasi data dengan waktu yang lebih singkat dibandingkan dengan algoritma konsensus PoW, sehingga mekanisme penyimpanan data dapat dilakukan secara langsung tanpa menunggu proses validasi. Hyperledger menyediakan beberapa *tool* yang mendukung impelementasi penelitian ini seperti Hyperledger Fabric yang dapat digunakan untuk membuat *blockchain* dan Hyperledger Composer yang dapat dimanfaatkan untuk menjalankan mekanisme penyimpanan data sensor IoT pada *blockchain*. Dengan menggunakan *platform* Hyperledger diharapkan sistem penyimpanan data sensor IoT dapat berjalan dengan menggunakan teknologi *blockchain.*

Sistem penyimpanan data sensor IoT akan dibuat dengan mendaftarkan dan memberikan hak akses kepada pengguna terlebih dahulu sebelum dapat melakukan mekanisme penyimpanan data pada *blockchain*. Mekanisme penyimpanan data pada *blockchain* dirancang dengan kemampuan untuk dapat menyimpan data ke dalam *blockchain*, mengubah data yang telah tersimpan dalam *blockchain*, dan mendapatkan data dari dalam *blockchain*.

2. LANDASAN KEPUSTAKAAN

Ayoade telah melakukan integrasi *blockchain* dengan IoT (Ayoade, et al., 2018). Penelitian Ayoade menerapkan *smart contract*

menggunakan Ethreum *blockchain* untuk mengatur hak akses dalam mengelola data IoT. Ayoade menyarankan menggunakan *permissioned blockchain* dengan waktu validasi yang lebih cepat.

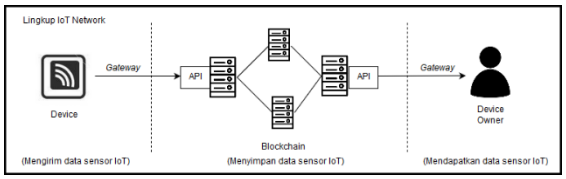
Yang, et al. (Yang, et al., 2018) mengimplementasikan pertukaran data pada *smart toy* menggunakan *platform* Hyperledger *blockchain* dengan tujuan untuk menjamin integritas dan konsistensi data.

Dogan & Seyrek membuat sistem IoT monitoring kendaraan dengan menerapkan *platform* Hyperledger Composer untuk menyimpan dan membuat *chaincode* pada *blockchain* (Dogan & Seyrek, 2018).

Christidis & Devetsikiotis (Christidis & Devetsikiotis, 2016) memberikan referensi mengenai *permissioned blockchain* dan *platform* hyperledeger *blockchain* untuk diterapkan pada implementasi yang akan dilakukan.

Androulaki, et al (Androulaki, et al., 2018) menjelaskan struktur komponen *blockchain* pada *platform* Hyperledger Fabric dan komponen-komponen yang terdapat dalam *platform* Hyperledger Fabric.

3. PERANCANGAN

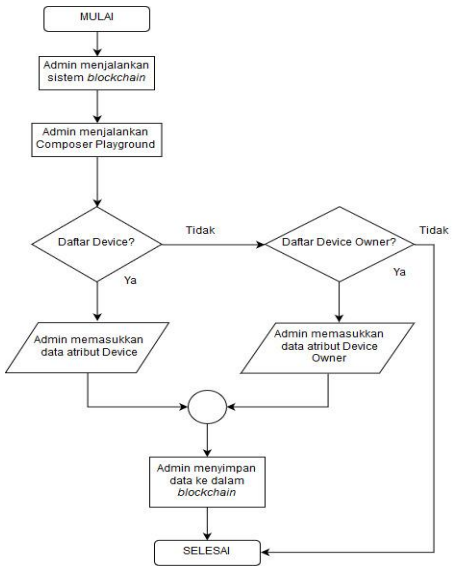


Gambar 1. Gambaran Umum Sistem

Gambar 1 memberikan gambaran umum mengenai lingkungan dari sistem penyimpanan data sensor IoT berbasis *blockchain.* Terdapat garis batasan untuk membatasi penelitian ini yang hanya berfokus dalam membuat sistem *blockchain* dan mekanisme penyimpanan data sensor IoT. Sehingga, sistem hanya menyediakan API yang dijalankan oleh *peer* node untuk menghubungkan *blockchain* dengan sistem lainnya. Terdapat dua entitas utama yang akan melakukan mekanisme penyimpanan data pada *blockchain* yaitu Device dan Device Owner. Device merupakan perangkat IoT yang akan menyimpan dan mengubah data sensor IoT. Sedangkan Device Owner adalah pengguna yang mengambil data sensor IoT dari *blockchain.*

3.1. Perancangan Mekanisme Pendaftaran Pengguna

Perancangan ini menggambarkan alur proses untuk mendaftarkan Device dan Device Owner ke dalam *blockchain* pada sistem penyimpanan data sensor IoT. Pendaftaran dapat dilakukan oleh Admin Blockchain ketika sistem *blockchain* telah berjalan. Gambar 2 menjelaskan diagram *flowchart* alur tahapan pendaftaran pengguna.



Gambar 2. Flowchart Pendaftaran Pengguna

Pendaftaran dilakukan dengan cara memasukkan atribut yang dimiliki oleh pengguna. Setelah mengisikan atribut pengguna, kemudian Admin Blockchain dapat menyimpan data pengguna tersebut. Apabila penyimpanan data oleh Admin berhasil dilakukan, maka pengguna akan langsung terdaftar ke dalam *blockchain* dan dapat dibuatkan *network card* sebagai identitas yang dapat digunakan untuk mengakses *blockchain. Network card* diberikan oleh Admin Blockchain kepada *peer* node Device atau Device Owner yang akan mengakses *blockchain.*

3.2. Perancangan Mekanisme Kontrol Akses Pengguna

Perancangan mekanisme kontrol akses pengguna berfungsi untuk mengatur batasan akses penyimpanan, perubahan, dan pengambilan data yang dapat dilakukan oleh pengguna *blockchain.* Mekanisme kontrol akses pengguna akan diatur dengan menggunakan *chaincode permission. Chaincode permission* berisi aturan-aturan yang dibuat untuk membatasi akses pengguna *blockchain. Chaincode Permission* dalam penelitian ini digunakan untuk mengatur kontrol akses Admin Blockchain, Device, dan Device Owner.

Mekanisme kontrol akses yang dapat dilakukan oleh pengguna terbagi menjadi tiga yaitu READ untuk melihat data, CREATE untuk memasukkan data, dan UPDATE untuk merubah data dalam *blockchain.* Pengaturan kontrol akses akan dibedakan berdasarkan tipe pengguna seperti pada Tabel 1.

Tabel 1. Kontrol Akses

Tipe Pengguna	Nama *Permission*	Keterangan
Admin Blockchain	AdmintoSyt em	AdmintoSystem mengatur izin hak akses untuk memperbolehkan Admin Blockchain mengakses sistem Hyperledger Composer dalam *blockchain.*
	AdmintoDat asensor	AdmintoDatasensor mengatur izin hak akses untuk melarang Admin Blockchain mengakses sistem data sensor IoT dalam *blockchain.*
	AdmintoDe viceowner	AdmintoDeviceowner mengatur izin hak akses Admin Blockchain untuk membuat dan melihat Device Owner dalam *blockchain.*
	AdmintoDe vice	AdmintoDevice mengatur izin hak akses Admin Blockchain untuk membuat dan melihat Device dalam *blockchain.*
Device	DevicetoSyt em	DevicetoSystem mengatur izin hak akses untuk memperbolehkan Device mengakses sistem Hyperledger Composer dalam *blockchain.*
	DevicetoDat asensor	DevicetoDatasensor mengatur izin hak akses Device untuk membuat dan merubah data sensor IoT dalam *blockchain.*
	DeviceToU pdateSensor Data	DeviceToUpdateSensor Data mengatur izin hak akses Device untuk merubah data sensor IoT melalui *chaincode* UpdateSensorData dalam *blockchain.*
Device Owner	Deviceowne rtoSytem	DeviceownertoSytem mengatur izin hak akses untuk memperbolehkan Device Owner mengakses sistem Hyperledger Composer dalam *blockchain.*
	Deviceowne rtoDatasens or	DeviceownertoDatasens or mengatur izin hak akses Device Owner untuk mengambil atau melihat data sensor IoT dalam *blockchain.*

3.3. Perancangan Mekanisme Penyimpanan Data

a. Perancangan Struktur Data

Perancangan ini berfungsi untuk menjelaskan bagaimana merepresentasikan data sensor IoT dan data pengguna dalam *blockchain.* Perancangan struktur data dilakukan dengan merancang *chaincode model*. *Chaincode* model terdiri dari beberapa bagian komponen seperti *asset*, *participant*, dan transaksi. *Asset* akan dirancang untuk merepresentasikan data sensor yang akan disimpan oleh pengguna dalam *blockchain.* Variable *asset* disimpan dalam method yang bernama Datasensor. Perancangan variable data *asset* dapat dilihat pada Gambar 3.

No.	Variable	Tipe	Keterangan
1.	`device`	Device	Identifier device pemilik data sensor
2.	`datasensorId`	String	Identifier data sensor
3.	`datasensor`	String	Representasi data sensor

Gambar 3. Variable Data *Asset*

Setelah mendifinisikan *asset* untuk data sensor, kemudian *participant* dirancang untuk merepresentasikan data pengguna yang memiliki data sensor. *Participant* dibuat sesuai dengan nama pengguna yang akan berpartisipasi pada *blockchain.* Nama pengguna dalam *blockchain* ini dibedakan menjadi dua yaitu Device dan Device Owner. Gambar 4 dan 5 menjelaskan perancangan variable yang dimiliki oleh Device dan Device Owner.

No.	Variable	Tipe	Keterangan
1.	`deviceId`	String	Identifier nama Device
2.	`namasensor`	String	Representasi nama Device yang akan memiliki data sensor

Gambar 4. Variable Data Device

No.	Variable	Tipe	Keterangan
1.	`datasensor`	Datasensor	Identifier data sensor yang akan dilihat oleh Device Owner
2.	`device`	Device	Identifier device pemilik data sensor yang dapat dilihat oleh Device Owner
3.	`deviceownerId`	String	Identifier dari nama device owner
4.	`namaowner`	String	Representasi nama Device Owner yang akan memiliki data sensor

Gambar 5. Variable Data Device Owner

Setelah merancang *participant* atau

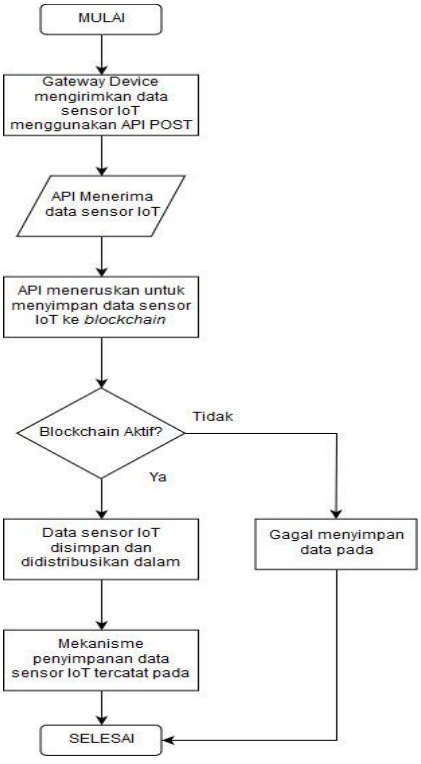
pengguna, kemudian transaksi dirancang untuk mengolah data yang dilakukan oleh pengguna dalam *blockchain.* Transaksi akan dimanfaatkan untuk mengupdate data sensor IoT. Transaksi dibuat dengan nama method UpdateSensorData. Variable yang akan dibuat pada model transaksi dapat dilihat pada Gambar 6.

No.	Variable	Tipe	Keterangan
1.	`newdatasensor`	String	Representasi data sensor baru
2.	`sampData`	Datasensor	Identifier data sensor untuk data sensor lama

Gambar 6. Variable Data Transaksi

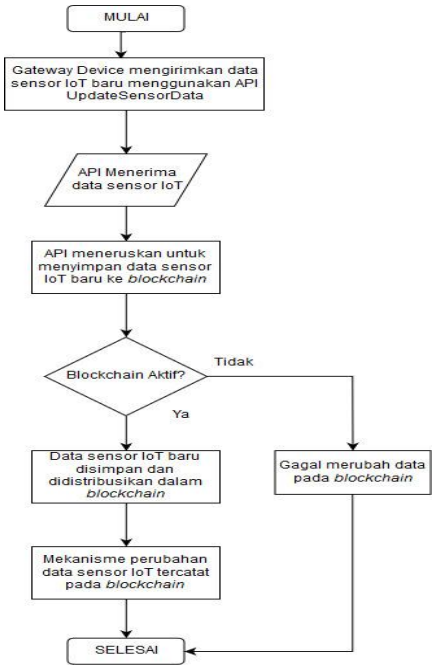
b. Perancangan Alur Mekanisme Penyimpanan Data

Perancangan alur mekanisme penyimpanan data pada *blockchain* terbagi menjadi tiga proses yaitu alur proses menyimpan data ke dalam *blockchain*, alur merubah data yang telah tersimpan dalam *blockchain*, dan alur mengambil data dari dalam *blockchain*. Gambar 7 menjelaskan diagram *flowchart* alur proses menyimpan data ke dalam *blockchain.*



Gambar 7. Alur Menyimpan Data ke Dalam *Blockchain*

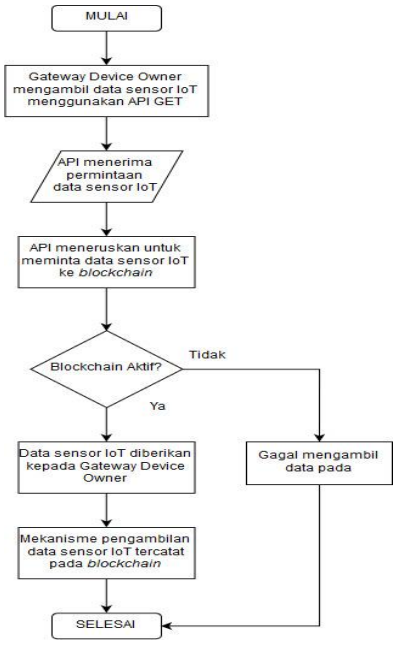
Alur menyimpan data ke dalam *blockchain* diawali dengan Gateway Device mengirimkan data yang mengandung payload data sensor IoT. Payload data sensor IoT berisi variable datasensorId yang menunjukkan Id dari data dan datasensor yang berisi nilai dari data sensor IoT. Pengiriman data dilakukan dengan menggunakan API POST dari Composer REST Server. Setelah API menerima data sensor IoT, kemudian data tersebut diteruskan untuk disimpan ke dalam *blockchain. Blockchain* akan menyebarkan data yang berhasil tersimpan kepada seluruh *peer* node dalam *blockchain.* Setelah tersimpan dan tersebar dalam *blockchain*, proses menyimpan data yang telah dilakukan akan tercatat oleh mekanisme pencatatan data pada *blockchain.* Selanjutnya, alur merubah data yang telah tersimpan dalam *blockchain* dijelaskan melalui diagram *flowchart* pada Gambar 8.



Gambar 8. Alur Merubah Data ke Dalam *Blockchain*

Perubahan data sensor IoT dapat dilakukan setelah terdapat data sensor IoT yang telah tersimpan dalam *blockchain.* Alur perubahan data ke dalam *blockchain* diawali dengan Gateway Device mengirimkan data sensor IoT baru menggunakan API UpdateSensorData. Payload data sensor IoT terdiri dari variable datasensorId yang berisi Id dari data yang mau dirubah dan variable datasensor yang berisi nilai data sensor IoT baru. Pengiriman data dilakukan dengan menggunakan API UpdateSensorData yang terbuat dari *chaincode transaction processor.* Setelah API menerima data sensor IoT, kemudian data tersebut diteruskan untuk disimpan ke dalam *blockchain. Blockchain* akan menyebarkan data yang berhasil tersimpan kepada seluruh *peer* node dalam *blockchain.* Setelah tersimpan dan tersebar dalam

blockchain, proses perubahan data sensor IoT yang telah dilakukan akan tercatat oleh mekanisme pencatatan data pada *blockchain.* Selanjutnya, alur mengambil data dari dalam *blockchain* dijelaskan melalui diagram *flowchart* pada Gambar 9.



Gambar 9. Alur Mengambil Data Dari Dalam *Blockchain*

Pengambilan data sensor IoT pada *blockhchain* dapat dilakukan setelah terdapat data sensor IoT yang telah tersimpan dalam *blockchain.* Alur pengambilan data pada *blockchain* diawali dengan Gateway Device Owner mengirimkan payload variable data sensor IoT yang berisi Id Device dan Id data sensor IoT yang akan diambil. Pengambilan data tersebut dilakukan dengan menggunakan API GET pada *blockchain.* Setelah API menerima permintaan data sensor IoT, kemudian diteruskan ke *blockchain.* Setelah *blockchain* berhasil memberikan permintaan data tersebut, proses pengambilan data sensor IoT yang telah dilakukan akan tercatat oleh mekanisme pencatatan data pada *blockchain.*

3.3. Perancangan Pengujian

Pengujian bertujuan untuk mengetahui apakah hasil implementasi sistem *blockchain* dapat berjalan sesuai dengan apa yang diharapkan pada kebutuhan dan perancangan sebelumnya. Pengujian akan dibagi menjadi dua bagian meliputi:

1. **Pengujian fungsional**

   Pengujian ini berfungsi untuk menguji fungsionalitas sistem berdasarkan dengan skenario pengujian fungsional. Pengujian ini akan memastikan semua fungsionalitas sistem dapat terpenuhi.

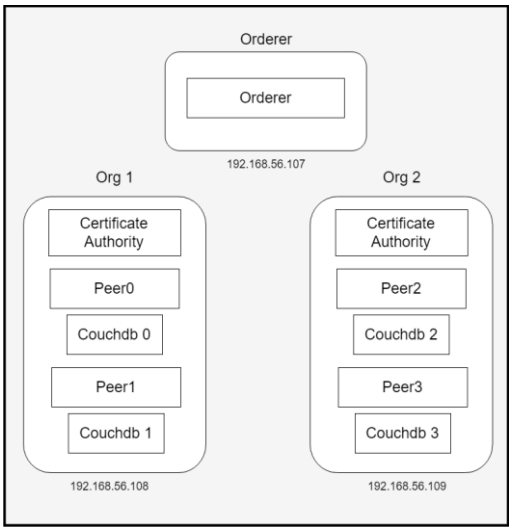
2. **Pengujian Non Fungsional**

   Pengujian non fungsional dilakukan untuk menguji kebutuhan non fungsionalitas sistem. Pengujian ini berfungsi untuk mengetahui keberhasilan sistem dalam menjaga privasi data pengguna, ketersediaan data, dan kinerja yang dapat dilakukan oleh sistem penyimpanan data sensor IoT berbasis *blockchain*.

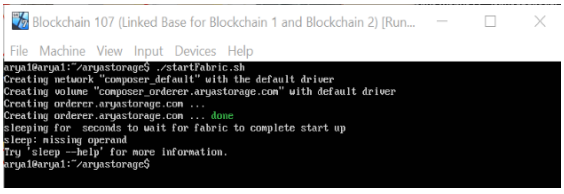
4. IMPLEMENTASI

4.1. Implementasi Mekanisme Pendaftaran Pengguna

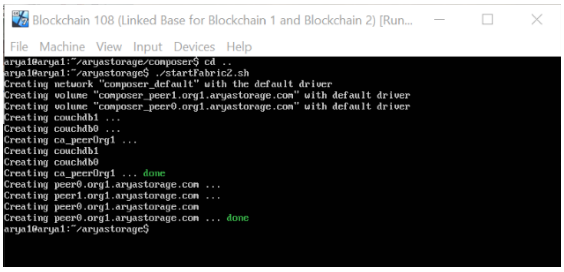
Implementasi mekanisme pendaftaran pengguna pada sistem penyimpanan data sensor IoT terbagi menjadi dua bagian yaitu menjalankan sistem *blockchain* dan pembuatan *network card. Blockchain* dijalankan degan tiga buah mesin virtual dengan IP mesin pertama 192.168.56.107, mesin kedua 192.168.56.108, dan mesin ketiga 192.168.56.109. Mesin virtual yang pertama dengan IP 192.168.56.107 menjalankan node yang berperan sebgai node Orderer Peer. Kemudian, mesin virtual kedua dengan IP 192.168.56.108 akan menjalankan node yang berperan sebagai node peer0 dan node peer1 dalam organisasi pertama. Selanjutnya, Mesin virtual ketiga dengan IP 192.168.56.109 menjalankan node yang berperan sebagai node peer2 dan node peer3 dalam organisasi kedua. Topologi dari ketiga mesin virtual pada *blockchain* dapat dilihat pada Gambar 10.



Gambar 10. Topologi *Blockchain*



Gambar 11. Menjalankan Mesin Virtual Pertama



Gambar 12. Menjalankan Mesin Virtual Kedua

Gambar 13. Menjalankan Mesin Virtual Ketiga

Gambar 11, 12 dan 13 menunjukkan *blockchain* berhasil dijalankan pada ketiga mesin virtual. Setelah menjalankan *blockchain*, pendaftaran pengguna dapat dilakukan dengan memasukkan atribut pengguna dan membuat *network card.* Gambar 14 menjelaskan cara membuat *network card* Admin Blockchain.

```
# create card for alice
$ composer card create -p ./byfn-network-org1.json -u alice -
n bnastorage -c alice/admin-pub.pem -k alice/admin-priv.pem
$ composer card import -f alice@bnastorage.card

# create card for bob
$ composer card create -p ./byfn-network-org2.json -u bob -n
bnastorage -c bob/admin-pub.pem -k bob/admin-priv.pem
$ composer card import -f bob@bnastorage.card

# create card for tito
$ composer identity issue -c cristi@bnastorage2 -f
tito@bnastorage2.card -u tito -a
"resource:org.aryastorage.arya.Deviceowner#333"
```

Gambar 14. *Network Card* Admin Blockchain

```
# create card for sensor suhu
$ composer identity issue -c alice -f
sensorsuhu@bnastorage.card -u SensorSuhu -a
"resource:org.aryastorage.arya.Device#001"

$ composer card import -f sensorsuhu@bnastorage.card

# create card for sensor jarak
$ composer identity issue -c bob -f
sensorjarak@bnastorage.card -u SensorJarak -a
"resource:org.aryastorage.arya.Device#002"

$ composer card import -f sensorjarak@bnastorage.card

# create card for sensor jarak
$ composer identity issue -c cristi@bnastorage2 -f
sensorjarak@bnastorage2.card -u sensorjarak -a
"resource:org.aryastorage.arya.Device#003"

$ composer card import -f sensorjarak@bnastorage2.card
```

Gambar 15. *Network Card* Device

```
# create card for Arya
$ composer identity issue -c alice -f arya@bnastorage.card -u
Arya -a "resource:org.aryastorage.arya.Deviceowner#111"
$ composer card import -f arya@bnastorage.card

# create card for Chandra
$ composer identity issue -c bob -f chandra@bnastorage.card -
u Chandra -a "resource:org.aryastorage.arya.Deviceowner#222"
$ composer card import -f chandra@bnastorage.card
```

Gambar 16. *Network Card* Device Owner

Kemudian, Gambar 15 dan 16 menjelaskan cara membuat *network card* untuk Device dan Device Owner dalam *blockchain.*

4.2. Implementasi Mekanisme Kontrol Akses Pengguna

Mekanisme kontrol akses pengguna dalam *blockchain* dilakukan dengan pengaturan akses pengguna pada *chaincode permission. Chaincode permission* dibuatkan kepada Admin Blockchain, Device, dan Device Owner untuk melakukan pengaturan mekanisme penyimpanan data yang berupa batasan yang dapat dilakukan pengguna untuk menyimpan, merubah, dan mengambil data sensor IoT pada *blockchain*.

No.	permissions.acl
1	/*======= Admin ========*/
2	rule AdmintoSystem {
3	description: "Allow Admin to access system resources"
4	participant: "org.hyperledger.composer.system.NetworkAdmin"
5	operation: ALL
6	resource: "org.hyperledger.composer.system.**"
7	action: ALLOW
8	}
9	rule AdmintoDatasensor {
10	description: "Allow Admin to access Datasensor resources"
11	participant: "org.hyperledger.composer.system.NetworkAdmin"
12	operation: ALL
13	resource: "org.aryastorage.arya.Datasensor"
14	action: DENY
15	}
16	rule AdmintoDeviceowner {
17	description: "Allow Admin to access Deviceowner resources"
18	participant: "org.hyperledger.composer.system.NetworkAdmin"
19	operation: CREATE, READ
20	resource: "org.aryastorage.arya.Deviceowner"
21	action: ALLOW
22	}
23	rule AdmintoDevice {
24	description: "Allow Admin to access Daevice resources"
25	participant: "org.hyperledger.composer.system.NetworkAdmin"
26	operation: CREATE,READ
27	resource: "org.aryastorage.arya.Device"
28	action: ALLOW
29	}
30	/*======= Device ========*/
31	rule DeviceToSystem {
32	description: "Allow Device to access system resources"
33	participant: "org.aryastorage.arya.Device"
34	operation: ALL
35	resource: "org.hyperledger.composer.system.**"
36	action: ALLOW
37	}
38	rule DeviceToDatasensor {
39	description: "Allow Device to Create and Update datasensor"
40	participant: "org.aryastorage.arya.Device"
41	operation: CREATE, UPDATE
42	resource: "org.aryastorage.arya.Datasensor"
43	action: ALLOW
44	}
45	rule DeviceToUpdateSensorData {
46	description: "Allow Device to UpdateSensorData"
47	participant: "org.aryastorage.arya.Device"
48	operation: CREATE
49	resource: "org.aryastorage.arya.UpdateSensorData"
50	action: ALLOW
51	}
52	/*======= DeviceOwner ========*/
53	rule DeviceownerToSystem {
54	description: "Allow Deviceowner to access system resources"
55	participant: "org.aryastorage.arya.Deviceowner"
56	operation: ALL
57	resource: "org.hyperledger.composer.system.**"
58	action: ALLOW
59	}
60	rule DeviceownerToDatasensor {
61	description: "Allow Deviceowner to Read datasensor"
62	participant(p): "org.aryastorage.arya.Deviceowner"
63	operation: READ
64	resource(r): "org.aryastorage.arya.Datasensor"
65	condition: (r.device.deviceId == p.device.deviceId)
66	action: ALLOW
67	}

Gambar 16. *Chaincode Permission*

Gambar 16 menujukkan *chaincode permission* yang mempunyai variable participant, operation, dan action. Variable participant berfungsi untuk menunjuk pengguna yang mendapatkan aturan. Kemudian, variable operation merupakan operasi mekanisme penyimpanan data yang dapat dilakukan pengguna tersebut seperti menyimpan (CREATE), merubah (UPDATE), dan mengambil (READ) data. Kemudian, variable action berfungsi menunjukkan apakah operasi mekanisme penyimpanan data diizinkan (ALLOW) atau tidak diizinkan (*DENY*).

4.3. Implementasi Mekanisme Penyimpanan Data

a. Implementasi Struktur Data

Penerapan struktur data dilakukan dengan membuat *chaincode* model. Gambar 17 menunjukkan *chaincode* model dalam penelitian ini.

```
No.  org.aryastorage.arya.cto
  1  namespace org.aryastorage.arya
  2
  3  asset Datasensor identified by datasensorId {
  4      --> Device device
  5      o String datasensorId
  6      o String datasensor
  7  }
  8
  9  participant Deviceowner identified by deviceownerId {
 10      --> Datasensor datasensor
 11      --> Device device
 12      o String deviceownerId
 13      o String namaowner
 14  }
 15
 16  participant Device identified by deviceId {
 17      o String deviceId
 18      o String namasensor
 19  }
 20
 21  transaction UpdateSensorData {
 22    --> Datasensor sampDevice
 23    o String newdatasensor
 24  }
```

Gambar 17. *Chaincode* Model

b. Menjalankan API *Blockchain*

Composer REST server digunakan untuk membuat sebuah API dari *blockchain.* Composer REST server dapat dijalankan perintah `composer-rest-server` pada salah satu mesin virtual. Gambar 18 menunjukkan contoh Composer REST Server yang berhasil dijalankan menggunakan *network card* Device pada mesin virtual dengan IP 192.168.56.109.

```
arya1@arya1:~/card$ composer-rest-server
? Enter the name of the business network card to use: sensorsuhu@bmastorage
? Specify if you want namespaces in the generated REST API: never use namespaces
? Specify if you want to use an API key to secure the REST API: No
? Specify if you want to enable authentication for the REST API using Passport: No
? Specify if you want to enable the explorer test interface: Yes
? Specify a key if you want to enable dynamic logging: n
? Specify if you want to enable event publication over WebSockets: Yes
? Specify if you want to enable TLS security for the REST API: No

To restart the REST server using the same options, issue the following command:
   composer-rest-server -c sensorsuhu@bmastorage -n never -u true -d n -u true

Discovering types from business network definition ...
Discovering the Returning Transactions..
Discovered types from business network definition
Generating schemas for all types in business network definition ...
Generated schemas for all types in business network definition
Adding schemas for all types to Loopback ...
Added schemas for all types to Loopback
Web server listening at: http://localhost:3000
Browse your REST API at http://localhost:3000/explorer
Rest Server dynamic logging is enabled
```

Gambar 18. Composer REST Server

5. PENGUJIAN

5.1. Pengujian Fungsional

Hasil dari pengujian fungsional yang telah dilakukan dapat dilihat pada Tabel 2. Tabel 2 menunjukkan seluruh kebutuhan fungsional telah berhasil berjalan sesuai dengan yang ditentukan.

Tabel 2. Hasil Pengujian Fungsional

No.	Kebutuhan Fungsional	Hasil Pengujian
1.	Sistem dapat mendaftarkan dan memberikan *network card* kepada Device dan Device Owner dalam *blockchain* melalui Admin Blockchain.	Valid
2.	Sistem dapat memberikan pengaturan kontrol akses kepada Device dan Device Owner dalam *blockchain* melalui Admin Blockchain.	Valid
3.	Sistem dapat menyimpan data sensor IoT yang dilakukan oleh Device dengan menggunakan API POST data.	Valid
4.	Sistem dapat merubah atau meng-*update* data sensor IoT yang dilakukan oleh Device dengan menggunakan API UpdateSensorData.	Valid
5.	Sistem dapat memberikan data sensor IoT kepada Device Owner dengan menggunakan API GET data.	Valid
6.	Sistem dapat mencatat dan memperlihatkan setiap pencatatan mekanisme penyimpanan dan perubahan data sensor IoT dalam *blockchain* kepada Device Owner.	Valid
7.	Sistem dapat mendistribusikan data yang disimpan oleh Device kepada seluruh *peer* node.	Valid

5.2. Pengujian Non Fungsional

1. Pengujian Privasi Data

Pengujian privasi data pengguna dilakukan dengan menggunakan kontrol akses dan *channel* dalam *blockchain*. Pengujian pertama dilakukan dengan menguji kontrol akses yang telah dibuat pada *chaincode permission.* Tabel 3 menunjukkan hasil pengujian kontrol akses berhasil berjalan sesuai dengan batasan akses yang telah dirancang pada Tabel 1.

Tabel 3. Hasil Pengujian Kontrol Akses

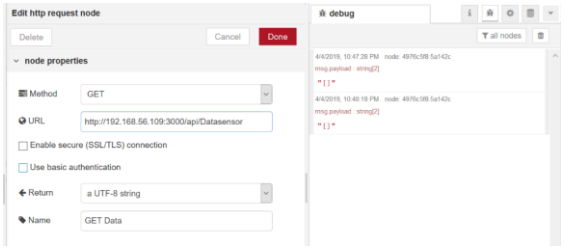
No.	Skenario Pengujian	Hasil Pengujian
1.	Admin Blockchain mencoba menyimpan data sensor IoT dengan API POST menggunakan *network card* Admin Blockchain.	Tidak Berhasil
2.	Admin Blockchain mencoba merubah data sensor IoT dengan API UpdateSensorData menggunakan *network card* Admin Blockchain.	Tidak Berhasil
3.	Admin Blockchain mencoba mengambil data sensor IoT dengan API GET menggunakan *network card* Admin Blockchain.	Tidak Berhasil
4.	Device Owner mencoba menyimpan data sensor IoT dengan API POST menggunakan *network card* Device Owner.	Tidak Berhasil
5.	Device Owner mencoba merubah data sensor IoT dengan API UpdateSensorData menggunakan *network card* Device Owner.	Tidak Berhasil
6.	Device mencoba mengambil data sensor IoT dengan API GET menggunakan *network card* Device.	Tidak Berhasil

Hasil pengujian pada Tabel 3 menunjukkan kontrol akses telah berhasil membatasi akses Admin Blockchain, Device, dan Device Owner. Kemudian, pengujian kedua dilakukan dengan menguji privasi dengan *channel* yang telah dibuat. Tabel 4 menunjukkan skenario pengujian privasi dengan *channel*.

Tabel 4. Pengujian *channel*

No.	Skenario Pengujian
1.	Menyimpan data sensor IoT melalui Device pada *channel* dalam organisasi dua.
2.	Device Owner pada organisasi satu dengan channel yang berbeda mencoba untuk mengambil data sensor IoT yang telah disimpan oleh Device pada organisasi dua.
3.	Device Owner pada organisasi dua dengan channel yang sama mencoba untuk mengambil data sensor IoT yang telah disimpan oleh Device pada organisasi dua

Hasil pengujian pada skenario Tabel 4 adalah Device Owner pada organisasi satu tidak dapat mengambil data sensor IoT yang berada pada organisasi dua karena berbeda *channel*. Data sensor IoT hanya dapat diambil oleh Device Owner pada organisasi dua dengan *channel* yang sama. Hal tersebut membuktikan *channel* pada organisasi dua berhasil membatasi akses pengambilan yang dilakukan oleh Device Owner dalam organisasi satu, sehingga privasi data pengguna terjaga dengan *channel* yang sama. Gambar 19 memperlihatkan hasil pengambilan data dengan *channel* yang berbeda.



Gambar 19. Privasi data dengan *channel*

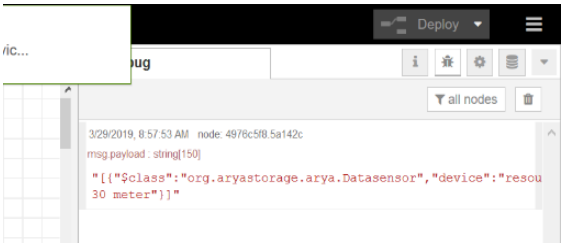
2. Pengujian Ketersediaan Data

Pengujian ketersediaan data menguji apakah pendistribusian data pada seluruh peer node yang dilakukan oleh sistem *blockchain* menjadikan data tersebar dan selalu tersedia pada seluruh peer node dalam *blockchain*. Tabel 5 menunjukkan skenario pengujian ketersediaan data.

Tabel 5. Pengujian Ketersediaan Data

No.	Skenario Pengujian
1.	Sistem blockchain telah berjalan.
2.	Salah satu mesin virtual dimatikan untuk memberhentikan *peer* node dalam blockchain.
3.	Device Owner mengambil data melalui *peer* node pada mesin virtual lain yang aktif dengan menggunakan API GET data.

Hasil pengujian ketersediaan data pada Tabel 5 adalah data sensor IoT dapat diambil oleh Device Owner melalui *peer* node pada mesin virtual yang berbeda. Kemudian, Device Owner dapat mengambil data sensor IoT dari peer node pada mesin virtual lain. Gambar 20 menunjukkan hasil data yang berhasil diambil melalui mesin virtual lain.



Gambar 20. Hasil Pengambilan Data

3. Pengujian Kinerja

Pengujian kinerja bertujuan untuk

mendapatkan waktu yang dibutuhkan dalam melakukan penyimpan, perubahan, dan pengambilan data pada blockchain. Pengujian dilakukan dengan menggunakan *tool* JMeter. Hasil pengujian kinerja dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian Kinerja

Proses	Hasil Pengujian
Penyimpanan data sensor IoT ke dalam blockchain	Latency dengan 10 data = 3015.1 ms Latency dengan 20 data = 5116.2 ms Latency dengan 30 data = 6732.433 ms Latency dengan 40 data = 7944.475 ms Latency dengan 50 data = 8721.66 ms
Perubahan data sensor IoT ke dalam blockchain	Latency = 2616 ms
Pengambilan data sensor IoT ke dalam blockchain	Latency = 368 ms

6. KESIMPULAN

Pada penelitian ini jenis pengguna dalam *blockchain* terbagi menjadi tiga tipe pengguna yaitu Admin Blockchain, Device, dan Device Owner. Mekanisme pendaftaran pengguna dalam *blockchain* dapat diterapkan dengan memasukkan data pengguna ke dalam *blockchain* melalui composer playground yang dilakukan oleh Admin Blockchain. Device dan Device Owner dapat melakukan mekanisme penyimpanan seperti menyimpan, merubah, dan mengambil data sensor IoT melalui API pada *blockchain. Blockchain* dapat menjaga privasi data pengguna dengan menggunakan kontrol akses pengguna dan *channel* dalam *blockchain.* Namun, aspek keamanan dan integritas data sensor IoT dalam *blockchain* tidak diuji dalam penelitian ini. Sehingga, perlu untuk dilakukan penelitian lebih lanjut terkait hal tersebut.

DAFTAR REFERENSI

Aazam, M., Khan, I., Alsaffar, A. A. & Huh4, E.-N., 2014. Cloud of Things: Integrating Internet of Things and Cloud Computing and the Issues Involved. Islamabad, IEEE, pp. 43-50.

Androulaki, E., Christidis, K. & Ferris, C., 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. Porto, Portugal, s.n.

Ayoade, G., Karande, V., Khan, L. & Hamlen, K., 2018. Decentralized IoT Data Management Using BlockChain and Trusted Execution Environment. dallas, s.n.

Cachin, C., 2016. Architecture of the Hyperledger Blockchain Fabric. IBM Research.

Christidis, K. & Devetsikiotis, M., 2016. Blockchains and Smart Contracts for the Internet of Things. SPECIAL SECTION ON THE PLETHORA OF RESEARCH IN INTERNET OF THINGS (IoT), Volume 4, pp. 2292-2303.

Composer, H., 2018. Welcome to Hyperledger Composer. [Online] Available at: https://hyperledger.github.io/composer

Dogan, M. & Seyrek, A., 2018. Vehicle Monitoring and Interaction to Blockchain, istanbul: Bogazici University.

Farooq, M. et al., 2015. A Review on Internet of Things (IOT). International Journal of Computer Applications, 113(1).

Foundation, A. S., n.d. Apache JMeter™. [Online] Available at: http://jmeter.apache.org/ [Accessed Saturday April 2019].

Hyperledger, C., 2017. Hyperledger Fabric Functionalities. [Online] Available at: https://hyperledger-fabric.readthedocs.io/en/release-1.2/functionalities.html#identity-management

Liu, B. et al., 2017. Blockchain Based Data Integrity Service Framework For IoT Data. New South Wales, IEEE.

Makhdoom, I., Abolhasan, M., Abbas, H. & Ni, W., 2018. Blockchain's adoption in IoT: The challenges, and a way forward. Journal of Network and Computer Applications.

Milenkovic, M., 2015. A Case for Interoperable IoT Sensor Data and Meta-data Formats. Issue The Internet of Things, p. 6.

Nick, S., 1994. Smart Contracts. [Online] Available at: http://szabo.best.vwh.net/smart.contracts.html

Porambage, P. et al., 2016. The Quest for Internet of Things. IEEE Cloud

Computing, Issue Privacy IoT, p. 37.

Rahman, A. F. A., Daud, M. & Mohamad, M. Z., 2016. Securing Sensor to Cloud Ecosystem using Internet of Things (IoT) Security Framework.

Yang, J., Lu, Z. & Wu, J., 2018. Smart-Toy-Edge-Computing-oriented Data Exchange Based on Blockchain. Journal of Systems Architecture.