

Kajian Literatur Penerapan Teknologi *Blockchain* pada *Human Computer Interaction*

Ega Rudy Graha¹ dan Ire Puspa Wardhani²

¹Mercedes Benz Mobility AG Siemensstr. 7, 70469 Stuttgart, Germany

²Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K

Jl. BRI Radio Dalam No. 17, Jakarta Selatan

E-mail: ega@graha.de, irepuspa@gmail.com

Abstrak

Teknologi *blockchain* digunakan untuk memastikan transaksi digital agar tidak dapat dimodifikasi dan dipalsukan. Platform aplikasi *blockchain* tersebar dengan topologi *cloud* ataupun non-*cloud*. *Multi-cloud computing* menjadi salah satu solusi untuk mengurangi ketergantungan terhadap penyedia *cloud* yang mendukung optimalisasi kinerja aplikasi dan efisiensi biaya. Interoperabilitas dan efisiensi penyimpanan data dan blok data dari *blockchain* dapat dipakai sebagai tolok ukur untuk menentukan sejauh mana sistem yang beragam dapat berintegrasi dan berjalan secara berirama. Salah satu cara untuk mengatasi permasalahan efisiensi penyimpanan data *blockchain* adalah dengan menerapkan standarisasi di semua tingkatan arsitektur. Tujuan penelitian yaitu untuk mengevaluasi penerapan teknologi *blockchain* berbasis *multi-cloud*, dengan berdasar pada kajian literatur dengan fokus pada optimalisasi interoperabilitas multi-*cloud* dan efisiensi dari kehandalan dan keamanan data. Proses penelitian ini meninjau dari beberapa solusi *cloud*, *multi-cloud* dan *edge computing* dengan mengelompokkannya ke dalam tiga pendekatan sebagai parameter standarisasi, yaitu analisa lapisan arsitektur, lingkungan yang homogen dan heterogen dari penyedia *cloud*, serta lingkungan antar perangkat *edge* untuk mendapatkan tingkat efisiensi penyimpanan data dan blok data pada teknologi *blockchain*. Dengan menjabarkan metode pendekatan dan jenis interoperabilitas serta tingkat efisiensi penyimpanan data dan blok data dari *blockchain*, dapat mengidentifikasi dan mengukur nilai optimal pemakaian platform sesuai kebutuhan pengguna.

Kata kunci : *Human Computer Interaction, Blockchain, Cloud Computing, Cloud Interoperability, Edge Computing*

Pendahuluan

Blockchain adalah database publik yang terdistribusi menggunakan sumber daya komputer untuk membuat blok yang dihubungkan bersama untuk tujuan melakukan transaksi. Teknologi *blockchain* dapat digunakan untuk memastikan transaksi digital tidak dapat dimodifikasi dan tidak dapat dipalsukan. Perkembangan inovasi teknologi di bidang sistem informasi dan sistem digital semakin pesat, seiring pula dengan penggunaan perangkat *edge* yang semakin inovatif seperti perangkat *IoT*, *smartphone*, dan bahkan kendaraan mengubah cara kita terhubung ke dunia fisik [1] yang sangat berhubungan erat dengan penggunaan *Human Computer Interaction*. Perangkat *edge* adalah perangkat yang menyediakan titik masuk ke jaringan utama penyedia layanan dalam implementasinya dalam *Human Computer Interaction*. Perangkat *edge* ini memproduksi sejumlah besar data. Dengan meningkatnya volume data yang di-

hasilkan perangkat *edge*, berbagi data di antara perangkat *edge* memungkinkan data diproses secara lokal tanpa keterlibatan *cloud* atau otoritas terpusat lainnya. Meskipun aplikasi-aplikasi saat ini digunakan di seluruh dunia melalui Internet, banyak dari mereka juga memiliki potensi besar dalam skenario *edge* dalam skala kecil, misalnya, pemantauan lingkungan lokal, dan perdagangan data.

Cloud computing adalah penyediaan layanan komputasi yang terdiri dari *server*, tempat penyimpanan data, database, jaringan, perangkat lunak, analitik dan kecerdasan buatan melalui internet untuk memungkinkan inovasi yang lebih cepat, sumber daya yang fleksibel serta terukur secara ekonomi. Di dalam *cloud computing* pengguna akhir mengakses aplikasinya tanpa perlu mengetahui dimana aplikasinya ditempatkan ataupun bagaimana layanan tersebut dipersiapkan. Hal ini tidak terlepas dari perkembangan penyedia *cloud* yang semakin banyak menawarkan solusi-solusi

yang beragam dan penetrasi akses internet berkecepatan tinggi yang semakin luas penggunaannya [2].

Layanan yang ditawarkan semakin bervariasi termasuk sumber daya teknologi informasi dan komunikasi, mulai dari perangkat keras termasuk jaringan, perangkat lunak seperti *stream-processing system*, *data-mining frameworks* dan aplikasi seperti email, *video on demand*, jejaring sosial dan aplikasi-aplikasi berbasis blockchain. Jenis layanan dari penyedia cloud [3] terdiri dari:

- *Infrastructure as a Service (IaaS)*
- *Platform as a Service (PaaS)*
- *Software as a Service (SaaS)*

Bagi pengguna cloud dengan mempunyai fleksibilitas sesuai kebutuhan pengguna menjadi sangat menarik dan diminati untuk mengeksplorasi layanan-layanan aplikasi dari penyedia *cloud*. Parameter penting yang mendukung pertimbangan mengenai *cloud computing* seperti *scalability* yang tinggi dan *flexible* sesuai kebutuhan. Sumber daya cloud disimpan di pusat data yang besar, yang sering disebut sebagai *data farms* dan dioperasikan oleh perusahaan-perusahaan besar seperti Amazon, Google, Microsoft, Apple, Salesforce dan IBM [3]. Dengan pendekatan *cloud* secara standar, dirancang untuk dikelola oleh pusat data tunggal. Pendekatan semacam ini memiliki beberapa kekurangan, seperti ketersediaan layanan yang mengalami gangguan yang tidak terduga ataupun kemungkinan adanya kegagalan pusat data, hal ini dapat mengganggu ribuan pengguna aplikasi dalam mengakses layanan aplikasi di dalam *cloud*. Selain itu mengeksploitasi pusat data tunggal menjadi sangat terbatas untuk menyimpan dan memproses data secara aman terutama yang berbasis geolokasi. Hal ini sering ditemukan untuk data-data yang bersifat legal, transaksi keuangan dan termasuk dalam klasifikasi *confidential regional data protection*.

Kemajuan *cloud computing* memicu revolusi terhadap hosting scenario dan penyediaan layanan aplikasi atau dikenal dengan model *deployment* yang berbasis cloud [4], yaitu terdiri dari:

- *Public cloud*, disediakan oleh penyedia cloud dipergunakan secara komersial
- *Private cloud*, disediakan dan dipergunakan oleh lingkungan sendiri
- *Community cloud*, penyediaan cloud dalam organisasi untuk keperluan anggota
- *Hybrid cloud*, penggunaan *cloud* yang layanannya merupakan kombinasi dengan lebih dari dua *cloud* yang berbeda antara *public*, *private* dan *community cloud*.

Multi-cloud computing menjadi salah satu solusi untuk mengurangi ketergantungan terhadap penyedia *cloud*, memenuhi peraturan regional, mengoptimalkan kehandalan *performance* dan biaya [5].

Hal ini menjadi sebuah paradigma dimana aplikasi menggunakan beberapa layanan *cloud* yang *heterogen* secara bersamaan. Strategi ini telah banyak diadopsi di industri *cloud computing* untuk meningkatkan ketahanan, penanganan *recovery* dari bencana dan permasalahan lokasi geografis, serta memastikan *scalability* aplikasi *cloud* yang menjadi tidak terbatas [6]. Kejadian *cloud outages* telah membuktikan bahwa “*don’t place all your eggs in one basket*” berlaku juga untuk ekosistem *cloud* [7].

Teknologi *blockchain* saat ini dipergunakan secara luas dalam *cryptocurrency*. *Blockchain* terdiri dari rantai blok, yang merupakan catatan atau buku besar. Setiap blok berisi hasil hash dari blok sebelumnya untuk membentuk rantai. *Blockchain* memiliki banyak fitur keamanan pada sistem terdistribusi. Pertama, histori lengkap disimpan di seluruh jaringan memudahkan untuk memulihkan dan memverifikasi informasi blok yang diperoleh pengguna. Kedua, *blockchain* dirancang untuk tahan terhadap modifikasi catatannya. Di lingkungan antar perangkat *edge*, baik lingkungan dalam skala kecil maupun lingkungan *cloud* dan *multi-cloud* bertujuan untuk mencapai akses data yang handal dengan mengelola access control secara detil dalam skala mikro dan transaksi pembayaran mikro secara terdistribusi.

Enterprise Architecture memungkinkan *interoperabilitas* dengan mendefinisikan building blocks dengan standarisasi *interface* yang dapat diwujudkan oleh layanan yang berinteraksi tanpa perlu banyak upaya integrasi. Salah satu *Enterprise Architecture Frameworks* modern yang luas dipergunakan di industri adalah TOGAF (*The Open Group Architecture Framework*) yang dipergunakan pertama kali oleh Departemen Pertahanan Amerika Serikat pada tahun 1995 [8], mempunyai keunggulan pada perancangan organisasi perusahaan sehingga dapat diterapkan langsung pada software development dengan model deployment di dalam cloud secara fleksibel seperti implementasi *multi-cloud*. Ada tiga pilar TOGAF antara lain: *TOGAF Architecture Domain*, *TOGAF ADM (Architecture Development Method)* dan *TOGAF Continuum* [9]. *Building blocks* yang dimaksud dapat dipetakan dengan *TOGAF Architecture Domain* [4] yang terdiri dari:

Business Architecture, menerangkan struktur bisnis dan interaksi antar strategi bisnis, organisasi, fungsi dan proses bisnis serta kebutuhan informasi.

Information System Architecture, terdiri dari:

- *Data Architecture* yang menerangkan struktur dan interaksi data.
- *Application Architecture* yang menerangkan struktur dan interaksi antar aplikasi di dalam kelompok yang koheren.

Technology Architecture, menerangkan struktur dan interaksi antar platform layanan pada tingkat

logikal dan fisik pada komponen-komponen teknologi.

Masalah substansial yang dihadapi industri dalam sistem terdistribusi adalah:

- Perangkat *edge* memiliki kendala pada sumber daya terutama tingkat interoperabilitas yang rendah antar *cloud* dan *non-cloud* yang menjadi tantangan bagi pengguna untuk mengatur sumber daya yang beroperasi dalam mendistribusikannya ke *platform* lainnya. *Interoperabilitas* tidak hanya mentransfer beban kerja antar *cloud*, tetapi juga mendistribusikan aplikasi dan data serta keamanan data baik *intra-cloud* [10] maupun dalam pusat data tetapi dengan lingkungan yang heterogen [11].
- Kompleksitas dan terdapatnya kasus-kasus duplikasi data dari sistem *blockchain* yang membuatnya tidak memungkinkan untuk mendistribusikannya secara langsung ke lingkungan perangkat *edge* [1] sehingga diperlukan mekanisme *blockchain* secara efisien.

Tantangan desain sistem untuk mengatasi keterbatasan perangkat *edge* pada aplikasi berbasis *blockchain* dengan aspek *Interoperabilitas*:

1. Bagaimana menyimpan data secara optimal dalam jaringan.
2. Bagaimana manajemen pengaturan data saat perangkat bergerak melakukan mobilitas. Tantangan rumusan masalah dari aspek efisiensi dalam sistem yang terpercaya dengan menggunakan mekanisme *blockchain*:
3. Bagaimana agar mencapai kemampuan penyimpanan data dengan mekanisme blok secara terdistribusi, pada perangkat *edge* secara handal dan terpercaya.

Penelitian ini bertujuan untuk mengevaluasi penerapan teknologi *blockchain* berbasis *multi-cloud*, berdasarkan kajian literatur dengan fokus penelitian pada optimalisasi *interoperabilitas multi-cloud* dan efisiensi kehandalan dan keamanan data.

Metode Penelitian

Penelitian ini merupakan kajian literatur dengan mengeksplorasi penelitian berfokus pada penerapan teknologi *blockchain* dalam perangkat *edge* dengan aspek *human computer interaction*.

Proses penelitian ini meninjau solusi desain pada teknologi *blockchain* dengan melibatkan platform yang heterogen [1] dengan mengelompokkan ke dalam 3 (tiga) pendekatan pada Tabel 1 sebagai parameter untuk langkah standarisasi, yaitu dengan menganalisa tingkat interoperabilitas agar dapat melakukan optimalisasi arsitektur dan konfigurasi infrastruktur, serta menganalisa efisiensi

blockchain dalam mekanisme penyimpanan data pada perangkat *edge*.

Tabel 1: Pendekatan Metode Penelitian Kajian Literatur

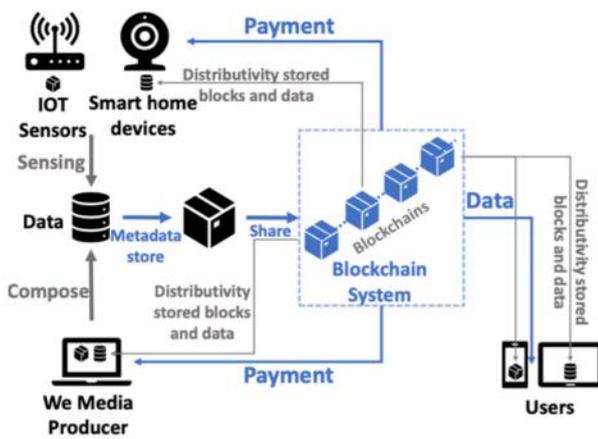
Pendekatan	Metode Penelitian	Sumber
1.	Analisa dari lapisan arsitektur, yaitu: <ul style="list-style-type: none"> • <i>3-tier architecture</i> • <i>Microservice architecture</i> 	[5] [7] [12] [13] [14] [15] [16]
2.	Analisa <i>multi-cloud</i> dalam lingkungan homogen dan heterogen.	[3] [10] [11] [17] [18]
3.	Analisa penerapan <i>blockchain</i> antar perangkat <i>edge</i> .	[1]

Pada masing-masing pendekatan metode penelitian, dilakukan analisa untuk tingkat efisiensi alokasi *blockchain* data dan penyimpanan blok data serta migrasi data *blockchain*.

Pembahasan

Teknologi *blockchain* diusulkan pada tahun 2008 oleh Satoshi Nakamoto dan telah banyak digunakan dalam *cryptocurrency* sejak saat itu. *Blockchain* terdiri dari serangkaian blok yang dihubungkan menggunakan kriptografi. Setiap blok biasanya berisi hasil hash dari blok sebelumnya, timestamp, hash untuk blok itu sendiri, dan beberapa item lain berdasarkan skenario aplikasi. Bersama-sama blok ini membentuk rantai. *Blockchain* dapat berfungsi sebagai buku besar terdistribusi untuk menyimpan data antar perangkat, dan data tidak dapat dengan mudah diubah karena fitur kriptografi. Jika pengguna yang ingin mengubah sepotong data, harus membuat seluruh rantai yang sangat rumit. Fitur anti-manipulasi *blockchain* dipergunakan sebagai dasar untuk implementasi *cryptocurrency*, misalnya, Bitcoin, Litecoin, dan Ethereum [1].

Sistem *blockchain* yang diusulkan pada literatur [1] terdiri dari beberapa perangkat *edge* yang menghasilkan transaksi data dan blok di *blockchain* (lihat Gambar 1). *Node* adalah perangkat *edge* yang berpartisipasi dalam sistem. Setiap *node* memiliki *private key* dan *public key* untuk tujuan identifikasi. *Private key* dan *public key* tersebut selanjutnya menghasilkan *account* dari simpul itu. Setiap *account* unik dan terkait dengan setiap *node* dan memiliki alamat unik yang memenuhi pola tertentu. Alamat *account* dapat dibuat dari *public key* tetapi tidak sebaliknya. Setiap *node* mencoba memalsukan blok untuk mendapatkan insentif, dan insentif tersebut akan disimpan di *account*-nya. Insentif ini diproduksi yang dikenal dengan nama token.



Gambar 1: Sistem landscape *blockchain* dalam aplikasi pembayaran yang diusulkan pada literatur [1]

Lapisan Arsitektur

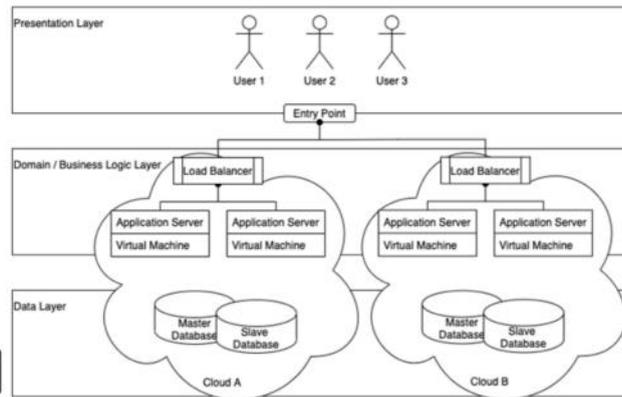
Membagi arsitektur ke dalam lapisan adalah pendekatan mendasar untuk memudahkan dalam pengelolaan kompleksitas. *Framework TOGAF* pilar *Architecture Domain* pada *Information System Architecture* [4], sistem dibangun dengan urutan lapisan yang masing-masing berdiri sendiri dan hanya dapat berkomunikasi dengan lapisan sebelumnya melalui *interface* yang telah didefinisikan. Pendekatan lapisan arsitektur ini memberikan kemudahan dalam pemahaman dan perawatan dimana lapisan-lapisan dapat dipersiapkan secara berdiri sendiri. Dalam referensi buku Fowler [19] dijelaskan tantangan terbesar dalam merancang sistem perangkat lunak *multi-tier* adalah bagaimana mengumpulkan komponen kepada masing-masing lapisan dan fungsi-fungsi apa saja yang dapat dikelompokkan. Kesalahan *design* dalam memutuskan pengelompokan lapisan ini akan membuat lapisan yang lebih kecil, akhirnya menyulitkan *interface* dalam berinteraksi.

3-tier Architecture

Standar aplikasi dengan lapisan arsitektur *cloud 3-tier* [7] [14] terdiri dari lapisan-lapisan sebagai berikut:

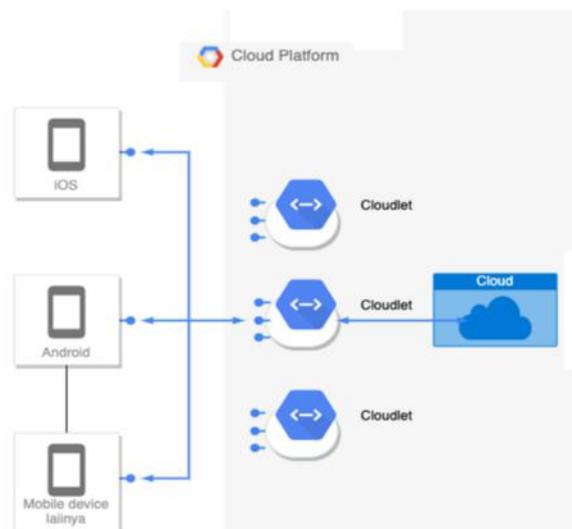
1. *Presentation Layer* – merepresentasikan *User Interface* dengan menampilkan kepada *user*.
2. *Domain / Business Logic Layer* – mengimplementasikan *Business Logic*.
3. *Data Layer* – menangani *Data Interface*, akses ke penyimpanan data.

Pada Gambar 2 merepresentasikan arsitektur *cloud 3-tier*:



Gambar 2: . *3-tier Cloud Architecture* [7]

Dari sudut pandang pengguna akhir *cloud* dapat dilihat secara hirarki terstruktur (lihat Gambar 3), yaitu pusat data terdiri dari host yang terbagi menjadi beberapa *virtual machine* dan *cloudlet* yang diarahkan ke *virtual machine* di dalam *cloud* tunggal maupun *multi-cloud*. *Cloudlet* yang dimaksud adalah pusat data *cloud* dengan ruang lingkup yang lebih kecil dan mempunyai karakteristik fungsional yang fleksibel dan terletak pada tepi jaringan internet dengan tujuan untuk mendukung aplikasi selular yang menyediakan sumber daya dengan kinerja tinggi dan latensi yang rendah. Latensi itu sendiri merupakan peristiwa penundaan sebelum transfer terjadi setelah instruksi transfer dilakukan. Interaksi pada tingkat *cloudlet* ini direalisasikan dalam bentuk migrasi data.



Gambar 3: Diagram hirarki pada *Cloudlet* [7]

Jenis Interoperabilitas

Dalam literatur arsitektur *cloud 3-tier* diungkapkan ada 2 (dua) jenis *Interoperabilitas* [7], yaitu:

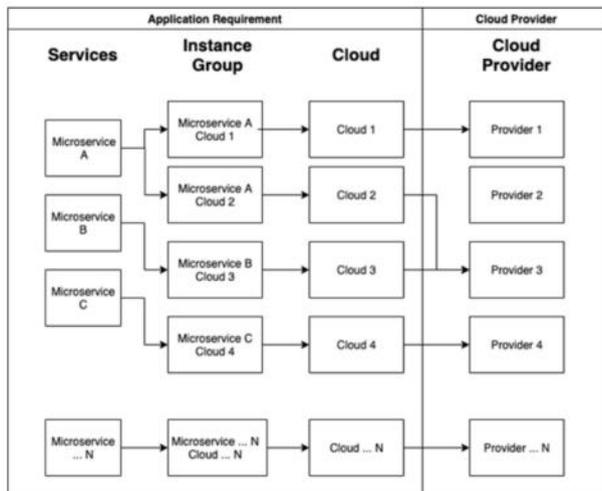
1. *Inter-Cloud Federation*, terdiri dari penyedia *cloud* utama yang berbeda. *Interoperabilitas* tinggi akan tercapai jika penyedia *cloud*

secara sukarela melakukan interkoneksi dan berbagi infrastruktur mereka.

2. *Multi-Cloud*, klien cloud mempunyai kemampuan untuk mendistribusikan beban kerja secara mandiri.

Efisiensi Alokasi *Blockchain* Data dan Penyimpanan Blok Data

Dalam lingkungan *Inter-Cloud Federation*, di mana tidak ada inter-operasi antara penyedia cloud utama yang bersaing seperti Amazon dan Google, pendekatan *Multi-Cloud* jauh lebih baik tingkat interoperabilitasnya daripada federasi dalam hal penyimpanan data. Dalam konfigurasi *Multi-Cloud*, interoperabilitas cloud tidak menjadi masalah, karena klien cloud tidak bergantung pada kolaborasi penyedia cloud. Masalah utama *Inter-Cloud Federation* adalah API manajemen yang berbeda dari semua penyedia *cloud* [7].



Gambar 4: Konfigurasi *Multi-Cloud* dengan arsitektur *microservice* [5]

Microservice Architecture

Microservice adalah desain arsitektur untuk membangun aplikasi terdistribusi menggunakan *container*. Setiap fungsi aplikasinya beroperasi sebagai layanan independen. Arsitektur ini memungkinkan layanannya untuk diperbaharui tanpa mengganggu layanan dari aplikasi lainnya. Arsitektur *microservice* menyediakan pendekatan untuk membangun aplikasi secara terukur, terdiri dari layanan kecil yang beroperasi dalam proses yang terpisah dan dapat dipergunakan di lingkungan yang berbeda. Selain keuntungan dari pemisahan secara modular, pendekatan ini mengukur layanan secara independen dan memungkinkan penggunaan sumber daya yang lebih baik. *Cloud* dengan *microservice* arsitektur memiliki pendekatan dengan pemodelan [5] [13] [15]:

1. *multi-cloud requirement*

2. *cloud provider variability*
3. *cloud provider mapping*

Dari ketiga pendekatan model arsitektur cloud untuk *microservice* tersebut direpresentasikan pada Gambar 4.

Container yang dimaksud merupakan *docker container* adalah berupa *image* paket perangkat lunak yang ringan, mandiri, dan dapat dieksekusi mencakup semua yang diperlukan untuk menjalankan aplikasi: *code*, *runtime*, *system device*, *system library*, dan pengaturan. *Instance Group* yang merupakan konfigurasi pasangan *microservice* pada suatu *cloud*, yang digambarkan dengan Gambar 4, diimplementasikan dengan *docker container* di dalam suatu cloud bertujuan untuk memudahkan pengembangan dan penyebaran perangkat lunak dengan meningkatkan daya portabilitasnya. Pengguna dapat dengan mudah membuat lingkungan yang disesuaikan dengan sempurna sesuai dengan kebutuhan produk perangkat lunaknya. Selain itu dapat menyebarkan replika lingkungan di dalam infrastruktur apa saja seperti *cloud*, *cluster*, dan di *level virtual machine*, dimana dapat diimplementasikan dengan cepat dibandingkan dalam menyiapkan sebuah *virtual machine* [15].

Jenis Interoperabilitas

Dalam literatur dijelaskan jenis-jenis interoperabilitas berdasarkan pengelompokan sebagai berikut [5]:

1. Kesamaan bahasa pemrograman untuk menentukan kelompok kriteria, yaitu *service requirement*, *cloud variable* dan *instance group*.
2. Model fitur untuk memodelkan kesamaan dan variabilitas di seluruh produk perangkat lunak atau sistem. Model fitur digambarkan sebagai diagram pohon yang nodenya mewakili fitur yang dapat dipilih untuk membangun produk perangkat lunak.
3. Ontologi untuk mengelola heterogenitas *cloud*, adalah definisi konsep formal dalam domain tertentu dan bagaimana mereka saling terkait. Hal ini didefinisikan oleh satu *set class* dan *object* yang terkait.

Efisiensi Alokasi *Blockchain* Data dan Penyimpanan Blok Data

Dalam lingkungan *microservice* dengan konfigurasi *multi-cloud*, mendefinisikan tingkat interoperabilitas dan efisiensi alokasi *blockchain* dan penyimpanan blok data dengan melakukan beberapa tahap pengujian, yaitu:

- Validasi keadaan *cloud*, misalnya, variabel *cloud A* memiliki syarat yang harus ditetapkan ke *cloud* yang terletak di Eropa Barat.

- *Mapping service requirement* terhadap penyedia *cloud*. Setelah memverifikasi bahwa penyedia yang ditetapkan telah mematuhi kondisi *cloud*, perlu memeriksa apakah setiap penyedia yang dipilih mendukung semua fungsionalitas yang diperlukan oleh layanan yang ditetapkan dengan menggunakan pemetaan penyedia *cloud*.
- Memverifikasi *feature constraints* dan menghasilkan konfigurasi lengkap. Setelah memetakan persyaratan layanan ke beberapa fitur, masih memungkinkan adanya batasan dalam sumber daya, fitur yang saling bertentangan, masih perlu memeriksa apakah semua layanan yang ditetapkan ke penyedia dapat digunakan bersama dan untuk menghasilkan konfigurasi untuk setiap penyedia.

Sebagai eksperimen dilakukan pengukuran terhadap tahapan pengujian tersebut dan menghasilkan tingkat operabilitas yang terukur, digambarkan pada Tabel 2.

Dalam literatur ini [5] pengujian dilakukan dengan 20 kali eksekusi untuk mengubah aplikasi ke konfigurasi multi-cloud. Dari hasil yang diperoleh, membutuhkan waktu rata-rata 6,62 detik. Untuk pengujian yang melibatkan 4 penyedia cloud, dengan 32 kombinasi solusi yang valid, secara total ada kemungkinan 1024 kombinasi assignment, dan ditemukan dalam waktu rata-rata 24 detik [5].

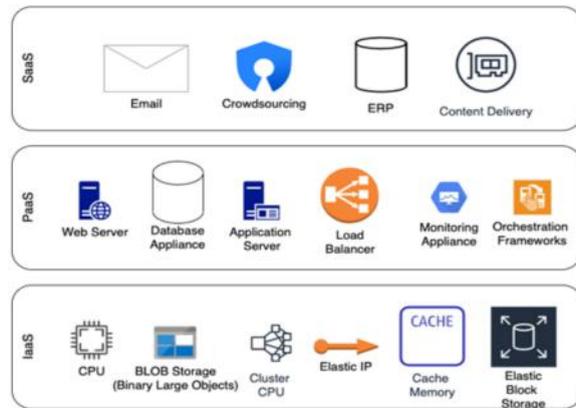
Tabel 2: . Nilai rata-rata interoperabilitas pada sebuah konfigurasi multi-cloud [5]

Operasi	Waktu rata-rata (ms)	Standard penyimpangan (ms)
Validasi keadaan <i>cloud</i>	3.56	5.12
Mapping <i>service requirement</i> terhadap penyedia <i>cloud</i>	1, 796.82	2, 215.69
Memverifikasi <i>feature constraints</i>	3, 350.47	827.34
Proses secara keseluruhan	6, 620.60	6, 431.53

Pengujian dapat dilakukan dengan kombinasi dari jumlah penyedia cloud yang bervariasi. Meskipun membutuhkan waktu untuk memproses dan menghasilkan solusi yang valid, assignment yang tidak valid dapat diabaikan dengan cepat dan mengurangi tahapan pencarian berikutnya [5]. Dari hasil dengan proses secara keseluruhan yang paling cepat bisa menjadi acuan sebagai tingkat interoperabilitas yang tinggi.

Multi-Cloud dengan Lingkungan Homogen dan Heterogen

Layanan *Cloud Computing* yang disediakan penyedia *Cloud* menggunakan informasi dan sumber daya teknologi seperti perangkat keras (CPU, GPU, memori dan jaringan), perangkat lunak (*database, streaming processing, dan data mining framework*) dan aplikasi (E-Mail, video) dan media sosial. Layanan ini disebut Infrastruktur sebagai Layanan (IaaS), Platform sebagai Layanan (PaaS), dan Perangkat Lunak sebagai Layanan (SaaS) [3]. Gambar 5 menunjukkan arsitektur berlapis dari model *cloud computing*.



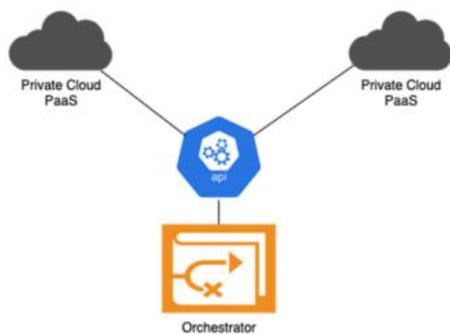
Gambar 5: Lapisan arsitektur di dalam *Cloud Computing* [3]

Dengan menghubungkan beberapa pusat data berbasis *cloud* atau *multi-clouds*, setiap pemilik aplikasi dapat meningkatkan kualitas layanan (QoS, *Quality of Service*), keandalan, dan fleksibilitas aplikasi mereka secara keseluruhan.

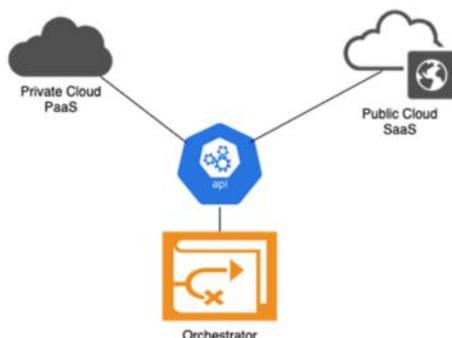
Jenis Interoperabilitas

Berdasar kajian literatur [3] pada layanan multi-clouds mempunyai jenis interoperabilitas yang terdiri dari :

1. *Multi-Cloud* homogen mempunyai jenis layanan *cloud* yang sama seperti IaaS – IaaS atau PaaS - PaaS, baik *Infrastructure as a Service* ataupun *Platform as a Service* [3] [11], digambarkan pada Gambar 6.
2. *Multi-Cloud* heterogen mempunyai jenis layanan *cloud* yang berbeda seperti IaaS - PaaS atau PaaS – SaaS, kedua kombinasi konfigurasi cloud dari jenis yang berbeda [11] [18], digambarkan pada Gambar 7.



Gambar 6: *Multi-cloud* dengan skema homogen *PaaS – PaaS* [3] [11]



Gambar 7: *Multi-cloud* dengan skema heterogen *PaaS – SaaS* [3] [11]

Dalam jenis *multi-cloud* heterogen mempunyai karakteristik alami di dalam pengambilan keputusan desain yang rumit, sebagai tantangan interoperabilitas yang tersedia dari beberapa penyedia *cloud*.

Efisiensi Alokasi *Blockchain* Data dan Penyimpanan Blok Data

Baik dalam lingkungan *multi-cloud* yang homogen maupun heterogen, kajian literatur telah mengidentifikasi penentuan tingkat interoperabilitas dan efisiensi alokasi *blockchain* dan penyimpanan blok data [3]. Tingkat interoperabilitas dan efisiensi alokasi *blockchain* dan penyimpanan blok data dilakukan dengan mengelompokkan tingkatannya sebagai berikut:

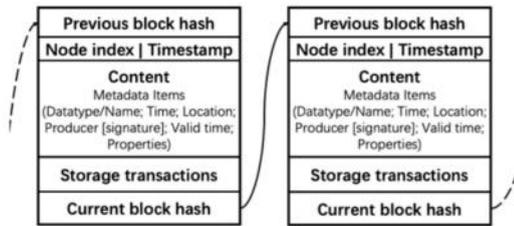
- Tingkat keragaman virtualisasi teknologi, yaitu dengan menghitung jumlah virtualisasi teknologi di dalam lingkungan *cloud*. Semakin besar jumlah virtualisasi teknologi menjadikan semakin tinggi tingkat interoperabilitas, hal ini disebabkan karena virtualisasi mengisolasi sumber daya perangkat keras, sehingga memungkinkan perilaku konteks keamanan yang toleran terhadap kesalahan dan terisolasi. Sehingga memungkinkan pemanfaatan yang lebih efisien dengan menyediakan fleksibilitas, *agility*, dan skalabilitas yang dibutuhkan untuk sumber daya fisik untuk mendukung *multitenancy* [3].

- Tingkat non-standard layanan dan biaya, serta SLA (*Service Level Agreement*) [3].
- Tingkat keragaman API, menentukan tingkat interoperabilitas yang meningkat jika jenis API yang dipergunakannya sejenis, tetapi akan menurunkan tingkat interoperabilitas jika jenis API yang dipergunakannya yang semakin beragam [3]. Perbedaan jenis API yang dipergunakan seperti SOAP yang dipergunakan untuk komunikasi *webservice* di aplikasi yang berbeda, sementara *REST* yang terintegrasi dengan *World Wide Web* dan tidak terbatas untuk komunikasi antar manusia tetapi dipergunakan untuk komunikasi antar mesin yang lebih sederhana [11].

- Tingkat non-standard teknologi untuk otentikasi dan otorisasi. Pada *private cloud* dan *public cloud* mempunyai mekanisme yang berbeda dalam memverifikasi pengguna untuk diizinkan memasuki area yang diinginkan dan diperbolehkan menggunakan data di area tersebut. Semakin tinggi tingkat non-standard yang dipergunakan, maka semakin rendah tingkat interoperabilitas dalam otentikasi dan otorisasi pengguna *cloud* [3].
- *Lock-in* penyedia *cloud*, salah satu yang menurunkan tingkat interoperabilitas, yang juga berarti ketergantungan yang tinggi terhadap penyedia *cloud* dan kurangnya kemampuan untuk memigrasikan komponen aplikasi dan mendistribusikan beban kerja terkait dari migrasi penyedia *cloud* A ke penyedia *cloud* B [3].

Penerapan *Blockchain* dengan Lingkungan Antar Perangkat *Edge*

Kebalikan dari *cloud computing* yang memindahkan komputasi ke *cloud* terpusat, *edge computing* memindahkan pekerjaan komputasi ke *node* terdistribusi di tepi jaringan. Komputasi sebagian besar atau seluruhnya terjadi pada *node* di dekat atau di dalam perangkat *edge* [1]. *Edge computing* dapat menawarkan kemampuan berbagi dan pemrosesan data yang cepat untuk perangkat akhir. Salah satu aspek penelitian utama dari *edge computing* mempelajari manfaat menggunakan server tepi yang lebih kecil (*cloudlet*) yang ditempatkan di dekat tepi jaringan (misalnya *cellular base station*), berfungsi sebagai lapisan tengah antara perangkat *edge* dan *cloud* [7]. Server tepi ini dapat menawarkan beberapa aplikasi seperti caching dan virtualisasi sumber daya.



Gambar 8: Usulan komponen blok dari sistem *blockchain* yang diusulkan literatur [1]

Gambar 8 menunjukkan komponen blok dalam sistem *blockchain* yang diusulkan. Dalam usulan sistem *blockchain* ini berisi informasi dasar dari *blockchain* yang khas, serta beberapa komponen yang dirancang untuk lingkungan perangkat *edge*. Hash sebelumnya, indeks, *timestamp*, dan *hash* saat ini seperti sistem *blockchain* biasa untuk memastikan koneksi antar blok. Isi di dalam blok menyimpan item metadata, masing-masing sesuai dengan item data. Selain itu, setiap blok mencatat informasi tentang di mana blok ini disimpan menggunakan transaksi penyimpanan.

Efisiensi Alokasi *Blockchain* Data dan Penyimpanan Blok Data

Pendekatan *edge computing* dengan teknologi *blockchain* memberikan penerapan yang lebih detail dan dibagi ke dalam kriteria berikut:

- Efisiensi lokasi penyimpanan
- Data dan alokasi penyimpanan blok
- Penyimpanan lokasi blok terbaru
- Data item dan proses akses blok
- Migrasi Data

Efisiens Lokasi Penyimpanan

Lingkungan *edge* tidak dapat menyimpan semua data di setiap node, data dan blok harus disimpan di tempat tertentu yang tersebar tetapi dapat diakses dengan mudah oleh pengguna yang membutuhkannya. Setiap node memiliki kapasitas yang berbeda, dan algoritma harus menyimpan lebih sedikit data pada node dengan sumber daya yang lebih sedikit [1].

Data dan Alokasi Penyimpanan Blok

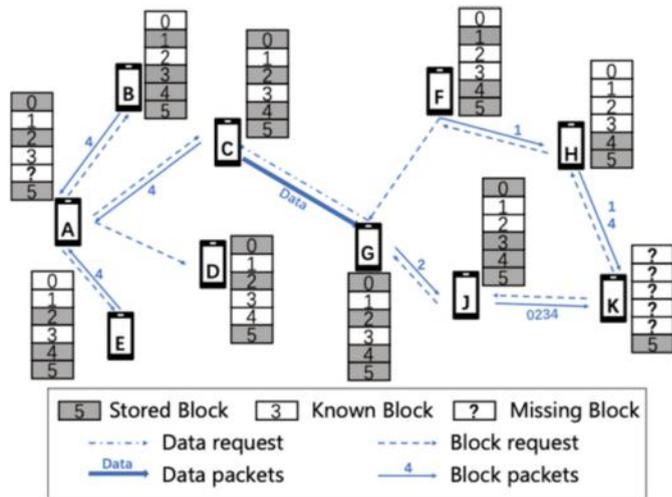
Ketika item data dihasilkan, produsen item data juga menghasilkan item metadata yang sesuai dan menyiarkannya. Setiap node yang menerima item metadata menghitung kumpulan node mana yang akan menyimpan item data [1].

Penyimpanan Lokasi Blok Terbaru

Mobilitas adalah salah satu karakteristik utama perangkat *edge*. Mobilitas node dapat menyebabkan koneksi yang tidak stabil, yang menyebabkan hilangnya data. Dengan demikian, blok *blockchain* terbaru adalah yang paling dibutuhkan untuk potensi pemutusan sementara node. Jika blok terbaru lebih meresap dalam jaringan, mengambil blok terbaru akan menjadi lebih mudah. Berbeda dari penyimpanan data dan blok, node diharuskan untuk men-cache sejumlah blok terbaru dan mengganti blok menggunakan FIFO [1].

Data Item dan Proses Akses Blok

Untuk *node* yang terputus dari jaringan dan membutuhkan blok terbaru, bisa mendapatkan blok dari node tetangga mana pun. Gambar 9 menunjukkan rute dalam mendapatkan akses ke blok yang diperlukan dari *node* tetangga.



Gambar 9: Proses akses blok komponen blok dari sistem *blockchain* yang diusulkan literatur [1]

Migrasi Data

Dalam lingkungan *edge*, bagaimanapun, mobilitas adalah salah satu karakteristik penting. Seiring waktu, node akan berubah lokasi, membawa item data yang tersimpan di dalamnya, dan informasi untuk mengakses data yang akan menjadi tidak optimal. Karena semakin banyak node bergerak dan mengubah topologi jaringan, keputusan penyimpanan harus disesuaikan terus menerus. Migrasi data akan memerlukan algoritma untuk menghitung dimana data akan disimpan jika topologi berubah.

Penutup

Hasil penelitian kajian literatur ini memperlihatkan,

1. Optimalisasi interoperabilitas dilakukan bertahap dengan memperhatikan:
 - (a) Arsitektur cloud yang dipergunakan oleh aplikasi, dimana n-tier arsitektur menjadi metode dalam mengidentifikasi jenis arsitektur cloud dan multi-cloud
 - (b) Heterogenitas cloud, adalah struktur dari lingkungan cloud untuk mengidentifikasi kemudahan interaksi antar cloud, semakin tinggi tingkat heterogenitas sebuah cloud akan semakin sulit sebuah cloud berinteraksi dengan cloud lainnya sehingga diperlukan standarisasi dalam penggunaan cloud.
2. Mekanisme penyebaran, penyimpanan dan migrasi data yang menjadi alat pendeteksi dalam mengidentifikasi efisiensi penyimpanan data dan blok dari aplikasi blockchain dalam menghasilkan blok data baru secara aman. Penelitian ini memberikan kesimpulan bahwa,
 - (a) Dengan mendefinisikan metode pendekatan dalam mencari nilai yang terbaik dalam interoperabilitas dan efisiensi penyimpanan blok data dari blockchain, dapat mengukur nilai yang terbaik dalam desain platform cloud yang sesuai dengan kebutuhan pengguna (optimal).
 - (b) Menggunakan teknologi blockchain menunjukkan tingkat keamanan data yang sangat terpercaya, ketersediaan data secara terdistribusi yang berkontribusi terhadap penghematan sumber daya sentral tetapi dapat memperoleh blok data baru yang sangat akurat. Penelitian Yang Akan Datang Penelitian yang akan datang perlu pendalaman dan implementasi dengan berfokus pada aspek interoperabilitas dari penggunaan teknologi standar terhadap tingkat optimalisasi interoperabilitas yang berkaitan dengan aplikasi blockchain pada edge computing yang terbukti efisien dan aman.

Daftar Pustaka

- [1] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye and Y. Yang, "Resource Allocation and Consensus of Blockchains in Pervasive Edge Computing Environments", *IEEE Transactions on Mobile Computing*, Nr. IEEE restricted to Rutgers University, pp. 1233 - 1536, 15 May 2021.
- [2] A. B. Albuquerque, N. C. Mendonça, P. V. Bessera, A. Camara and R. Ximenes, "Cloud-step: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud", *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Nr. Programa de Pós-Graduação de Instituições de Ensino Particulares (PROSUP)*, p. 1, 2012.
- [3] R. Ranjan, "The Cloud Interoperability Challenge", *IEEE Cloud Computing*, Nr. the IEEE Computer Society, p. 20, 2014.
- [4] The Open Group, "Cloud Computing Portability and Interoperability", *Open Group Guide*, Nr. The Open Group, p. 9, 2013.
- [5] G. Sousa, W. Rudametkin and L. Duchien, "Automated Setup of Multi-Cloud Environments for Microservices Applications," *IEEE Computer Society*, Bd. 49, Nr. The IEEE Computer Society, p. 328, 2016.
- [6] S. Challita, F. Paraiso and P. Merle, "Towards Formal-based Semantic Interoperability in Multi-Clouds", *IEEE Computer Society*, Bd. 98, Nr. The IEEE Computer Society, p. 710, 2017.
- [7] N. Grozev and R. Buyya, "Performance Modelling and Simulation of Three-Tier Applications in Cloud and Multi-Cloud Environments", *The British Computer Society*, Bd. Vol. 58 No. 1, Nr. The Computer Journal, The British Computer Society, p. 1, 2013.
- [8] E. B. Setiawan, "Pemilihan EA Framework", *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2009)*, Nr. Seminar Nasional Aplikasi Teknologi Informasi, pp. B-115, 2009.
- [9] I. M. Edward, A. Agusdian, W. Shalannanda and S. I. Lestaringati, "Proposal of TOGAF ADM Enterprise Continuum for Organization-Specific Solution on e-Government", *IEEE International Conference on Electrical Engineering and Computer Science*, Nr. IEEE, p. 2, 2014.
- [10] G. Girish and N. J. Nischita "Cloud Broker and their role in a Hybrid Multi Cloud Environment", *IEEE*, Nr. IEEE, p. 1533, 2017.
- [11] S. M. Barhate and M. P. Dhore, "Hybrid Cloud: A Cost Optimised Solution To Cloud Interoperability", *International Conference on Innovative Trends in Information Technology (ICI-TIIT)*, Nr. IEEE Xplore, pp. 1 - 4, 23 May 2020.
- [12] C.-M. Chituc, "Towards a Methodology for Trade-off Analysis in a Multi-Cloud Environment Considering Monitored QoS Metrics and Economics Performance Assessment Results", *IEEE International Conference on Cloud Computing Technology and Science*, Bd. 7th, Nr. IEEE International Conference on Cloud Computing Technology and Science, p. 479, 2015.

- [13] D. Elliott, C. Oteroq, M. Ridley and X. Merino, "A Cloud-Agnostic Container Orchestrator for Improving Interoperability", International Conference on Cloud Computing, Bd. 11th, Nr. The IEEE Computer Society, p. 958, 2018.
- [14] M. A. Al Zain, E. Pardede, B. Soh and J. A. Thom, "Cloud Computing Security: From Single to Multi-Clouds", Hawaii International Conference on System Sciences, Bd. 45th, Nr. The IEEE Computer Society, p. 5491, 2012.
- [15] M. AbdelBaky, M. Unuvar and M. Steinder, "Docker Containers Across Multiple Clouds and Data Centers", International Conference on Utility and Cloud Computing, Bd. 8th, Nr. IEEE Computer Society, pp. 368, 369, 370, 2015.
- [16] A. Sampaio and N. Mendonça, "Uni4Cloud: An Approach based on Open Standards for Deployment and Management of Multi-cloud Applications", SE-CLOUD'11, Bd. 11, Nr. SE-CLOUD'11, p. 16, 22 May 2011.
- [17] S. M. Barhate and M. P. Dhore, "Hybrid Cloud: A Solution To Cloud Interoperability", Proceeding of International Conference on Inventive Communication and Computational Technologies (ICICCT 2018), Bd. 2nd, Nr. IEEE Xplore , p. 1242, 2018.
- [18] E. Majda and E. Ahmed, "Using Cloud SaaS to ensure interoperability and standardization in heterogeneous Cloud based environment", IEEE, Nr. IEEE, pp. 1, 3, 4 - 6, 2015.
- [19] M. Fowler, "Patterns of Enterprise Application Architecture", New York, NY, USA: Addison-Wesley, 5 Nov. 2002.