

## **GEO-REPLICATION DALAM TINJAUAN LATENSI DAN EFEKTIFITAS BIAYA**

**Taufiq O. D. Putra<sup>1)</sup>, Adi S. S. Ansyah<sup>2)</sup>, Miftahol Arifin<sup>3)</sup>, dan Royyana M. Ijtihadie<sup>4)</sup>**

<sup>1, 2, 3, 4)</sup> Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

e-mail: [taufiq.odhi@gmail.com](mailto:taufiq.odhi@gmail.com)<sup>1)</sup>, [adisur.sa@gmail.com](mailto:adisur.sa@gmail.com)<sup>2)</sup>, [miftaholarifin2017@gmail.com](mailto:miftaholarifin2017@gmail.com)<sup>3)</sup>, [roy@its.ac.id](mailto:roy@its.ac.id)<sup>4)</sup>

### **ABSTRAK**

*Replikasi adalah teknik pendistribusian data untuk sinkronisasi antar database agar data tetap konsisten. Replikasi dapat mengatasi masalah kehilangan data dan melakukan pemulihan sistem dengan cepat jika terjadi masalah pada salah satu server. Salah satu masalahnya yaitu ketika terjadi bencana alam di lokasi server. Akibatnya jika tidak mempunyai replikasi data yang berbeda lokasi maka akan menyebabkan sistem tidak berjalan dan kemungkinan kehilangan data. Kemudian, geo-replikasi dapat mengurangi latensi karena jarak client dengan data center jauh lebih dekat. Penerapan geo-replication pada umumnya mereplikasi data ke semua data center. Akibatnya, biaya penerapannya tinggi karena membutuhkan banyak resource. Karena berbagai kelebihan dan kekurangan dalam penerapannya, diperlukan pengelompokan teknik teknik geo-replication untuk memudahkan peneliti dan teknisi untuk menyesuaikan sesuai kebutuhan. Makalah ini menyurvei teknik Geo-replication untuk penerapan efektifitas biaya dan latensi. Artikel yang disurvei meliputi metode pemilihan letak replikasi, metode pengurangan round trip time, metode sesuai tipe data, dan pemilihan leader untuk menentukan server node yang digunakan. Hasil dari survey artikel menunjukkan bahwa penerapan Geo-replication untuk efektifitas biaya lebih cocok untuk digunakan pada sistem yang semua usernya tidak perlu mengakses seluruh data. Sedangkan untuk latensi rendah lebih cocok untuk sistem yang digunakan berbagai jenis user. Makalah ini dapat memanfaatkan teknik teknik yang sudah di tinjau untuk mengatasi masalah efektifitas biaya dan latensi dalam penerapan Geo-replication.*

**Kata Kunci:** *Geo-replication, Low Latency, Cost Effective*

## **GEO-REPLICATION IN A REVIEW OF LATENCY AND COST-EFFECTIVENESS**

**Taufiq O. D. Putra<sup>1)</sup>, Adi S. S. Ansyah<sup>2)</sup>, Miftahol Arifin<sup>3)</sup>, dan Royyana M. Ijtihadie<sup>4)</sup>**

<sup>1, 2, 3, 4)</sup> Department of Informatics, Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

e-mail: [taufiq.odhi@gmail.com](mailto:taufiq.odhi@gmail.com)<sup>1)</sup>, [adisur.sa@gmail.com](mailto:adisur.sa@gmail.com)<sup>2)</sup>, [miftaholarifin2017@gmail.com](mailto:miftaholarifin2017@gmail.com)<sup>3)</sup>, [roy@its.ac.id](mailto:roy@its.ac.id)<sup>4)</sup>

### **ABSTRACT**

*Replication is a data distribution technique for synchronization between databases so that data remains consistent. Replication can overcome data loss problems and perform system recovery quickly if a problem occurs on one of the servers. One of the problems is when a natural disaster occurs at the server location. As a result, if you do not have data replication in different locations, it will cause the system to not run and possibly lose data. Then, geo-replication can reduce latency because the distance between the client and the data center is much closer. The application of geo-replication in general replicates data in all data centers. As a result, the cost of implementation is high because it requires a lot of resources. Because of the various advantages and disadvantages in its application, it is necessary to group geo-replication techniques to make it easier for researchers and technicians to adjust as needed. Therefore, this paper surveys the articles on Geo-replication techniques to implement cost-effectiveness and latency. The articles surveyed included a method for selecting replication sites, a method for reducing round trip time, a method according to data type, and selecting a leader to determine which server node to use. The results of the article survey show that implementing geo-replication for cost-effectiveness is more suitable for use in systems where all users do not need to access all data. Meanwhile, low latency is more suitable for systems used by various types of users. This paper can utilize the techniques that have been reviewed to overcome the problem of cost-effectiveness and latency in implementing Geo-replication.*

**Keywords:** *Geo-replication, Low Latency, Cost Effective.*

### **I. PENDAHULUAN**

**D**I era digital terjadi peningkatan jumlah pengguna internet. Sehingga untuk mendapatkan pengalaman pengguna yang baik diperlukan teknik untuk memberikan layanan yang cepat dan sedikit terjadi kesalahan. Teknik replikasi data berperan dalam pendistribusian data untuk ketersediaan data yang tinggi dan toleransi dalam kegagalan sistem. Replikasi dalam sistem terdistribusi mengacu pada distribusi data dari server ke server lain sambil terus memperbarui dan menyinkronkan data dengan sumber sehingga pengguna dapat mengakses data yang relevan dengan aktivitas mereka [1].

Teknik replikasi memiliki 2 model, yaitu model pertama replikasi yang terletak dalam satu lokasi *data center* dan model kedua replikasi yang tersebar di beberapa *data center* dengan lokasi geografis berbeda. Masing masing model memiliki kekurangan dan kelebihan sesuai dengan kebutuhan. Replikasi model pertama lebih cocok untuk yang tidak mempunyai banyak pengguna tersebar di belahan dunia dan untuk menghemat biaya. Walaupun demikian model pertama memiliki kekurangan ketersediaan data secara global yang mengakibatkan latensi tinggi jika memiliki client di berbagai lokasi karena jarak *Data center* (DC) dan *client* yang jauh [2]. Selain itu, dapat mengakibatkan kemungkinan kehilangan data akibat dari bencana alam. Teknik replikasi dengan menyebar penyimpanan data di berbagai lokasi geografis disebut dengan *Geo-replication* [3].

*Geo-replication* pada umumnya bekerja dengan cara memperbarui data dari lokasi utama yang kemudian di replikasi ke semua lokasi penyimpanan DC [4]. Saat ini, *Geo-replication* mudah direalisasikan menggunakan layanan *cloud* seperti Amazon Web Service, Google *Cloud Platform*, Azure dan layanan *cloud* lainnya. Meskipun *geo-replication* dapat diimplementasikan dengan mudah, cara untuk mendapatkan performa terbaik menggunakan penerapan replika yang optimal masih belum jelas.

Performa penerapan *geo-replication* bergantung pada beberapa faktor, seperti jarak antar DC yang mempengaruhi *Round Trip Time* (RTT), jarak *client* dengan DC yang mempengaruhi latensi dan pemilihan *leader* untuk menentukan *node* server yang dipakai [2][5][6]. Pada penerapan *geo-replikasi* untuk mendapatkan latensi rendah perlu mereplikasi data ke seluruh DC yang tersebar di berbagai lokasi. Karena jika DC lebih dekat dengan *client* maka waktu yang diperlukan untuk memproses *request client* dapat dipersingkat [7]. Akan tetapi, penerapan tersebut membuat biaya yang tinggi dikarenakan penggunaan resource penyimpanan data yang banyak karena data direplikasi di semua DC. oleh karena itu perlu diketahui terlebih dahulu kebutuhan sistem yang lebih mementingkan latensi rendah atau efisiensi biaya dalam penerapan teknik *Geo-replication*.

Karena replikasi data akan memainkan peran yang semakin penting di *cloud*, tujuan dari makalah ini adalah untuk mensurvei teknik replikasi pada sistem *Geo-replication*. Sepengetahuan kami, makalah survei ini adalah upaya pertama untuk memeriksa masalah replikasi data pada sistem *Geo-replication* secara komprehensif dan sistematis dengan fokus khusus pada sistem *Geo-replication*.

Kontribusi kami pada artikel ini yaitu mengelompokkan metode–metode replikasi pada sistem terdistribusi *geo-replication*, yaitu metode replikasi yang dilakukan berdasarkan letak lokasi *node – node* server yang berbeda-beda. Letak lokasi *node* server ini bukan hanya berbeda antar Gedung, atau antar kota, namun bisa berada pada negara yang berbeda, sampai di benua yang berbeda, sehingga memiliki kondisi lokasi yang berbeda–beda pula. Pengelompokkan metode-metode replikasi pada sistem *ge-replication* ini bertujuan untuk memudahkan para peneliti atau teknisi di bidang teknologi informasi untuk memelihara data baik untuk instansi mereka atau untuk kebutuhan pribadi mereka karena pada saat ini, banyak instansi yang memerlukan data mereka aman atau ingin layanan yang mereka buat dapat digunakan oleh tidak hanya masyarakat di sekitar mereka, melainkan orang-orang di seluruh dunia. Salah satu cara agar memberikan layanan yang terbaik untuk orang-orang di seluruh dunia adalah mendekatkan layanan tersebut ke *user*.

Dalam teknik *geo-replication* dalam lingkungan terdistribusi sering kita melihat pembahasan para peneliti eksperimen terkait tujuan teknik tersebut mengenai masalah efisiensi biaya dan latensi rendah. Dari dua masalah tersebut kami mengelompokkan beberapa metodologi teknik *geo-replication* dalam menurunkan biaya dan menghasilkan latensi yang rendah. Dalam efisiensi biaya kami membagi metodologi tersebut berdasarkan tipe data yang ada serta tata letak *node* yang digunakan dalam teknik *geo-replication*. Sementara untuk metodologi dalam menurunkan latensi dalam *geo-replication* data kami berfokus pada proses replikasi data ke berbagai server *node* ini agar lebih efisien, cepat, dan dapat mempertahankan konsistensi data pada masing-masing server *node*.

Makalah ini bertujuan untuk membahas teknik *geo-replication* untuk menangani efektifitas biaya dan latensi rendah serta variannya. Contohnya termasuk penerapan *geo replication* dengan fokus pada jenis tipe data, penempatan *Data center*, dan latensi. Hasil dari makalah ini, peneliti dan praktisi dapat memanfaatkan teknik teknik yang sudah di tinjau untuk mengatasi masalah efektifitas biaya dan latensi sesuai dengan kebutuhan dalam penerapan *geo-replication*.

Bagian selanjutnya akan disusun dengan bagian 2 yang memaparkan penelitian terkait. Bagian 3, kami menjelaskan detail methodology dan prosedur penelitian ini. Bagian 4 terdapat hasil dan pembahasan yang berisi rangkuman teknik teknik untuk mewujudkan penyelesaian masalah yang diusulkan dalam penerapan *geo-replication*. Bab terakhir berisi kesimpulan dari penelitian yang sudah dilakukan.

## II. PENELITIAN TERKAIT

### A. *Service replication taxonomy in distributed environments*

Mengelompokkan metode replikasi data pada sistem terdistribusi dengan studi kasus *Service-Oriented Architecture* (SOA), *cloud*, dan *mobile* [8]. Artikel ini juga menjelaskan secara umum kelebihan, kekurangan, tantangan, dan penelitian selanjutnya terkait metode replikasi data pada SOA, *cloud*, dan *mobile*. Metode –

metode replikasi dikelompokkan berdasarkan tipe replikasi dan siapa yang mengelola replikasi. Untuk tipe replikasi ada dua, yaitu *static replication* dan *dynamic replication*. Pada metode *dynamic replication* dikelompokkan lagi menjadi dua, yaitu *dynamic placement* dan *dynamic selection*. Pengelompokkan berdasarkan siapa yang mengelola replikasi dibagi menjadi *distributed client management*, *centralized server management*, dan *distributed server management*.

Metode-metode replikasi juga diklasifikasi dengan beberapa bentuk. Pertama, yaitu aktif, pasif, dan semi aktif. Klasifikasi kedua yaitu, paralel, serial, dan komposit. Ketiga mengklasifikasikan menjadi kombinasi antara aktif, pasif, dan waktu. Kemudian klasifikasi yang keempat mengelompokkan berdasarkan *Directed Acyclic Graph* (DAG). Klasifikasi yang terakhir mengelompokkan menjadi, *active*, *passive*, *dynamic parallel*, dan *dynamic sequential*. Pada makalah kami mengelompokkan metode replikasi pada sistem terdistribusi pada kasus yang berbeda yaitu pada studi kasus *Geo-replication* dan mengklasifikasikan dengan bentuk yang berbeda yaitu *cost* efektif dan latensi rendah.

#### B. Survey on replication techniques for distributed system

Menjelaskan beberapa metode replikasi pada sistem terdistribusi, diantaranya *Read One Write All* (ROWA), *Quorum Consensus* (QC), *Tree Quorum* (TQ) *Protocol*, *Grid Configuration* (GC) *Protocol*, *Two-Replica Distribution Techniques* (TRDT), *Neighbor Replica Triangular Grid* (NRTG), dan *Neighbor Replication Distributed Techniques* (NRDT) [9]. Artikel ini juga menjelaskan terdapat dua tipe metode replikasi yaitu metode secara *synchronous* dan *asynchronous*. Pada artikel ini tidak menjelaskan metode-metode replikasi diimplementasikan pada studi kasus tertentu, hanya menjelaskan secara singkat beberapa metode replikasi pada sistem terdistribusi, sedangkan pada makalah kami mensurvei metode-metode replikasi pada studi kasus *Geo-replication*.

#### C. A comprehensive review of the data replication techniques in the cloud environments

Menjelaskan tentang teknik dan mekanisme replikasi data dalam lingkungan *cloud* [10]. Artikel ini juga memberikan gambaran taksonomi mekanisme replikasi data *cloud* yang akan di tinjau. Mekanisme replikasi data *cloud* dibagi menjadi dua bagian utama yaitu mekanisme statis dan dinamis. Mekanisme statis replikasi data yaitu penentuan lokasi *node* replikasi selama fase desain sedangkan mekanisme dinamis dalam replikasi merupakan pemilihan *node* replikasi pada saat dijalankan. Dalam penelitian ini kemudian dilakukan perbandingan mengenai mekanisme replikasi data yang diusulkan. Dalam strategi mekanisme statis, jumlah replica dan lokasi awalnya ditetapkan terlebih dahulu. Kemudian dalam strategi mekanisme dinamis secara dinamis membuat dan menghapus suatu replica sesuai dengan beban lingkungan yang ada. Tinjauan menunjukkan bahwa beberapa pendekatan dinamis memungkinkan strategi replikasi terkait mereka untuk disesuaikan pada waktu berjalan sesuai dengan perubahan perilaku pengguna dan topologi jaringan. Dalam makalah ini dijelaskan teknik replikasi data dalam lingkungan *cloud* secara dinamis, dimana teknik replikasi dinamis dinilai dapat mengoptimalkan latensi. Teknik replikasi data dinamis memiliki keterkaitan dengan makalah survei yang kami sajikan yaitu *Geo-replication* data untuk mencapai latensi rendah.

#### D. A survey of data replication techniques for mobile ad hoc network databases

Menjelaskan tentang survei serta klasifikasi data untuk database Jaringan ad hoc seluler (MANET). Makalah ini mengidentifikasi isu-isu yang terlibat dalam replikasi data MANET dan upaya untuk mengklasifikasikan teknik replikasi data MANET yang ada berdasarkan isu-isu yang ditangani [11]. Kemudian atribut dari teknik replikasi juga ditabulasikan untuk memfasilitasi perbandingan fitur dari pekerjaan replikasi data MANET yang ada. Parameter dan metrik kinerja juga disajikan untuk mengukur kinerja teknik replikasi MANET. Selain itu, makalah ini juga mengusulkan kriteria pemilihan teknik replikasi data yang sesuai untuk berbagai kebutuhan aplikasi tidak ada replikasi data yang mengatasi semua masalah dalam MANET yaitu tidak satupun dari teknik tersebut yang *power-aware*, *partition-aware as well as real-time-aware*. Pengembangan teknik replikasi data yang mempertimbangkan ketiga masalah diharapkan menjadi topik yang menarik untuk penelitian selanjutnya. Pada makalah kami mengelompokkan metode replikasi pada sistem terdistribusi dalam studi kasus *Geo-replication* serta tinjauannya dalam efektifitas biaya dan latensi rendah.

#### E. Data replication schemes in cloud computing: a survey

Dalam penelitian ini menjelaskan target utama adanya replikasi data yaitu untuk mencapai kinerja yang lebih tinggi untuk aplikasi intensif data serta mengatasi masalah seperti ketersediaan, keandalan, *bandwidth*, keamanan serta waktu respon akses data yang cepat. Dalam makalah ini disajikan survei komprehensif dan klasifikasi skema replikasi data yang canggih dalam komputasi awan. Klasifikasi skema replikasi data yang disajikan terdiri dari tiga kelas utama yaitu skema audit data, deduplikasi data dan skema penanganan data [12]. Dalam artikel ini juga digambarkan perbandingan komparatif yang lengkap dari skema replikasi dimana perbandingan tersebut menyoroti berdasarkan faktor-faktor penting, seperti kelas yang digunakan, jenis skema, tempat implementasi,

alat evaluasi, serta kelebihan dan kekurangannya. Selama proses replikasi data dalam sistem *cloud*, terdapat beberapa tantangan yang sering dihadapi, seperti migrasi replikasi, penempatan replikasi, keamanan, replikasi multi lingkungan, toleransi kesalahan, dan skema replikasi. Pada makalah ini dijelaskan survei replikasi data dalam komputasi awan berdasarkan tiga kelas yang dijelaskan diatas. Sedangkan makalah yang kami ajukan yaitu survei metode-metode replikasi pada studi kasus *Geo-replication*.

#### F. A Systematic Literature Review of the Data Replication Techniques in the Cloud Environments

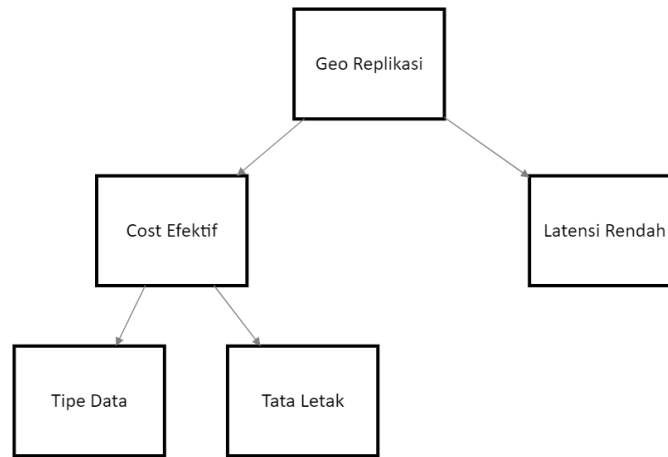
Replikasi data merupakan salah satu fenomena yang dipelajari paling luas dalam lingkungan terdistribusi dimana banyak Salinan dari beberapa data disimpan dalam banyak situs dimana biaya pembuatan, pembaruan replica dan pemeliharaan replika merupakan masalah yang penting untuk dibahas. Makalah ini menyajikan tinjauan sistematis terhadap mekanisme replikasi data masa lalu dan canggih dalam lingkungan *cloud*. Penelitian ini juga memperkenalkan taksonomi mekanisme replikasi data *cloud* yang ditinjau [13]. Replikasi data di lingkungan *cloud* terbagi dalam dua kategori metode statis dan dinamis. Dalam pola statis, sejumlah replika yang dibuat adalah konstan dan tetap sejak awal. Sedangkan , dalam algoritma dinamis dimana pembuatan replika mempertimbangkan lingkungannya, angka ditentukan berdasarkan algoritma akses pengguna. Tujuan dari makalah ini adalah untuk meninjau teknik replikasi data dalam dua kelompok utama secara sistematis serta membahas fitur utama dari masing-masing kelompok dimana menghasilkan teknik replikasi data dalam lingkungan *cloud* yang meningkatkan ketersediaan data, keandalan yang ditingkatkan, peningkatan kinerja serta toleransi kesalahan. Pada makalah yang kami usulkan yaitu mengelompokkan teknik-teknik replikasi dalam lingkungan terdistribusi dalam mencapai latensi rendah serta efektifitas biaya studi kasus *Geo – replication*.

Secara umum penelitian-penelitian terkait di atas mengklasifikasikan metode-metode replikasi pada sistem terdistribusi berdasarkan *platform* yang digunakan, terutama pada *cloud* dan *mobile*, walaupun beberapa ada yang mengelompokkannya secara lebih spesifik lagi. Beberapa diantaranya juga mengelompokkan berdasarkan protokol jaringan yang digunakan. Pada artikel ini akan mengklasifikasikan metode-metode replikasi pada sistem terdistribusi dengan pengelompokan yang berbeda, yaitu pada *Geo-replication*, dengan pengelompokan utama yaitu metode replikasi untuk *cost* efektif, dan latensi rendah.

### III. METODOLOGI

Makalah ini mengelompokkan beberapa metode replikasi pada *Geo-replication* yang tujuannya untuk mengoptimalkan *cost* atau biaya yang diperlukan untuk melakukan proses replikasi atau disebut juga dengan *cost* efektif dan memperkecil latensi. Pada metode replikasi untuk *cost* efektif beberapa ada yang menggunakan tipe data pada metode yang diusulkan dan ada yang berfokus pada tata letak atau penentuan letak dimana *node* server diletakkan.

Metode replikasi pada *Geo-replication* untuk *cost* efektif bertujuan tidak hanya menghemat *cost* atau biaya yang perlu dikeluarkan untuk proses replikasi data naum juga untuk membuat infrastruktur menjadi lebih dapat diandalkan atau *reliable* [14]. Di sisi lain, metode replikasi pada *Geo-replication* untuk latensi rendah bertujuan untuk memberikan layanan yang terbaik pada user dan memaksimalkan profit yang didapatkan, karena latensi sekecil apapun akan berpengaruh pada keberhasilan pada suatu instansi [15]. Untuk memudahkan pembuatan program untuk sistem terdistribusi *Geo-replication* maka tipe data khusus dikembangkan [7]. Menentukan tata letak *node* server dapat mempengaruhi seberapa luas jangkauan dari suatu sistem terdistribusi, dan hal ini dapat mempengaruhi kemampuan dari sistem tersebut untuk memenuhi tugasnya [16]. Pada Gambar. 1 menunjukkan penggambaran pohon taksonomi atau pengelompokkan metode replikasi pada *Geo-replication* yang dibuat pada artikel ini.



Gambar. 1. Pohon Taksonomi yang Diajukan

### A. Cost efektif

Proses replikasi data pada sistem *geo-replication* memerlukan biaya yang besar, karena sumber data bisa berada di beberapa tempat atau perlu direplikasi ke beberapa tempat. Oleh karena itu terdapat penelitian–penelitian yang mengajukan metode replikasi data pada sistem *geo-replication* dengan tujuan untuk mengefektifkan biaya. *Cost* efektif bukan berarti *cost* atau biaya yang dikeluarkan harus sedikit mungkin, namun *cost* atau biaya dikeluarkan pada saat yang tepat, dan proses replikasi tidak dilakukan secara redundan, atau dengan kata lain sesuai dengan kebutuhan.

#### 1) Tipe data

Salah satu komponen dari data adalah tipe data. Mereplikasi data pada suatu tipe data berbeda dengan tipe data yang lainnya. Oleh karena itu terdapat penelitian–penelitian untuk membuat suatu tipe data khusus yang bertujuan untuk mudah direplikasi sehingga dapat diimplementasikan pada sistem terdistribusi. Mudah direplikasi pada konteks ini bisa berupa berbagai bentuk, seperti menyederhanakan proses replikasinya dengan menghilangkan *proxy* atau pihak ketiga ketika terjadi konflik data atau memudahkan pada sisi pemrograman sehingga proses replikasi dapat dilakukan dengan mudah dan aman.

#### 2) Tata letak

Letak server *node* satu dengan server *node* lainnya merupakan salah satu yang mempengaruhi *cost* atau biaya yang diperlukan ketika melakukan proses replikasi. Semakin dekat jarak antar server *node* maka *cost* yang diperlukan akan lebih sedikit, begitu juga sebaliknya semakin jauh jarak antar server *node* maka *cost* yang diperlukan akan semakin besar. Oleh karena itu terdapat penelitian–penelitian yang mengajukan suatu pola tata letak server *node* pada sistem terdistribusi agar ketika melakukan proses replikasi data antara server *node* satu dan server *node* lainnya, tidak memerlukan *cost* yang terlalu besar atau *cost* yang dikeluarkan bisa seoptimal mungkin.

### B. Latensi rendah

Salah satu ciri dari sistem *geo-replication* adalah letak server *node* untuk menyimpan data dapat berada di tempat yang berbeda–beda. Oleh karena, itu diperlukan proses replikasi data ke server *node* lainnya, sehingga ketika *user* yang berada pada lokasi yang jauh dari tempat server *node* berada, *user* dapat mengakses data yang diinginkan melalui server *node* lainnya yang lebih dekat karena sama–sama memiliki data yang dibutuhkan oleh *user* dan memperkecil latensi ketika proses pengiriman data dari server ke *user* karena jaraknya lebih dekat. Penelitian–penelitian juga dilakukan untuk mengajukan metode bagaimana proses replikasi data ke berbagai server *node* ini bisa efisien, cepat, dan dapat mempertahankan konsistensi data pada masing–masing server *node* agar dapat digunakan secara optimal.

## IV. PEMBAHASAN DAN DISKUSI

Salah satu yang membedakan metode–metode *geo-replication* dengan metode replikasinya yaitu, metode *geo-replication* diajukan untuk melakukan replikasi data antara beberapa server *node* yang terletak di daerah yang berbeda-beda. Letak server *node* yang berbeda-beda ini sering kita temui pada sistem terdistribusi yang sekarang umum digunakan karena semakin meningkatnya kebutuhan komputasi dan penyimpanan data untuk menunjang teknologi-teknologi yang berkembang lainnya. Berikut akan dijelaskan beberapa metode *geo-replication* yang akan dikelompokkan menjadi dua kelompok utama yaitu, metode-metode *geo-replication* untuk *cost* efektif dan metode–metode *geo-replication* untuk latensi rendah.

### A. *Cost Efektif*

Metode – metode yang dikembangkan untuk *cost* efektif ini antara lain, pertama yaitu *Karma* merupakan sistem untuk replikasi data parsial atau sebagian, yang dapat mempertahankan *causal consistency* dan tanpa masalah terkait *availability* dan latensi [17]. Sistem *Karma* ini juga dapat menurunkan *cost* yang diperlukan ketika melakukan proses replikasi parsial dan tentunya memiliki *cost* yang lebih kecil daripada proses replikasi secara menyeluruh.

Pada penelitian lainnya mengatakan bahwa teknik replikasi penuh sudah tidak praktis lagi dalam pemrosesan banyak data dalam *Data Center* (DC). Salah satu cara untuk meningkatkan ketersediaan data dapat dilakukan dengan menggunakan replikasi antar DC sehingga data dapat diakses secara lokal, jika memungkinkan, yang memungkinkan untuk memulihkan jika ada kegagalan situs dan mengurangi biaya akses. Dengan demikian mereplikasi data hanya di beberapa DC menjadi lebih penting dalam mengurangi biaya untuk menjaga konsistensi data yang akhirnya konsisten [5], tetap mempertahankan ketersediaan tinggi dan biaya akses yang rendah. Lokasi data secara keseluruhan DC harus ditentukan secara dinamis memberikan pola perubahan permintaan baca dan tulis untuk data yang akan direplikasi.

Kemudian penelitian selanjutnya menjelaskan dalam mode penyimpanan *cloud*, untuk mengatasi kehilangan data dalam *cluster* digunakan teknologi replikasi acak tiga arah, dan *geo-replication* cluster umumnya digunakan untuk melindungi kegagalan terkait ini. Untuk mengurangi biaya lebih rendah dalam *geo-replication* cluster penuh dilakukan pengaturan praktis dengan menggunakan dua replica sudah cukup dalam melindungi dari kegagalan *node* independen. Replikasi tiga arah dianggap berlebihan dalam melindungi dari kegagalan *node* independen dan jelas tidak melindungi sistem penyimpanan dengan kegagalan *node* yang berkorelasi [18]. Replikasi berjenjang kemudian diusulkan sebagai solusi, yaitu model replikasi yang membagi cluster menjadi level master dan backup. Dua salinan pertama disimpan di bagian master dan digunakan untuk memulihkan data jika terjadi kegagalan *node* independen, sedangkan salinan ketiga disimpan di bagian cadangan dan digunakan untuk melindungi dari kegagalan terkait.

Penelitian selanjutnya yaitu mengusulkan dua metode yaitu *CostTLO* [19], metode yang mencegah latensi tinggi ketika mengakses data di beberapa server *node* sekaligus, dan *SPANStore* [20] yaitu metode yang membantu mengetahui pola latensi dan harga sehingga dapat menentukan *cost* secara efektif ketika melakukan proses replikasi data [21]. Penelitian yang mengusulkan metode *CostTLO* dan *SPANStore* ini juga mengarah ke metode replikasi data untuk latensi rendah, walaupun metode ini lebih dominan untuk *cost* efektif.

Pada penelitian lainnya juga menggunakan metode *SPANStore* pada kasus yang berbeda. Selama fase pengembangan, *SPANStore* menyediakan tampilan terpadu dari layanan penyimpanan yang didistribusikan secara geografis untuk aplikasi, mengotomatiskan proses pengalihan biaya dan latensi sambil memenuhi persyaratan konsistensi dan toleransi kesalahan. Desain *SPANStore* yang diajukan mencapai tujuan tersebut dengan menjangkau pusat data dari beberapa penyedia *cloud*, dengan cerdas menentukan kebijakan replikasi berdasarkan karakteristik beban kerja dan meminimalkan penggunaan sumber daya komputasi. Dalam hal ini dilakukan implementasi *SPANStore* di layanan *cloud* Amazon, Microsoft dan Google dan menemukan bahwa *SPANStore* dapat menawarkan keuntungan biaya yang signifikan dibandingkan praktik replikasi sederhana. Evaluasi terhadap *SPANStore* menunjukkan bahwa dalam beberapa skenario, *SPANStore* dapat mengurangi biaya lebih dari 10 kali lipat dibandingkan dengan solusi alternatif yang menggunakan satu ruang penyimpanan atau mereplikasi setiap objek ke setiap pusat data yang melaluinya *SPANStore* dapat diakses [20].

Selanjutnya, dalam mencapai latensi yang rendah serta biaya yang rendah dalam system *geo-replication* parsial yaitu dengan 3 tahapan, pertama diharapkan menggunakan atribut objek yang tepat dalam membuat keputusan replikasi untuk setiap jenis aplikasi. Kedua menggunakan replikasi multi atribut dan yang ketiga adalah menggunakan replikasi secara terus menerus. Dilakukan evaluasi terhadap ide tersebut dengan prototype Apache Cassandra dalam mengimplementasikan prototipe Acorn dan sistem *geo-replication* parsial berkesinambungan berbasis Atribut [22]. Eksperimen menunjukkan pengurangan biaya dan latensi yang cukup besar dengan dua jenis aplikasi berbagi data. Beberapa metode *geo-replication* untuk *cost* efektif menggunakan pendekatan tipe data dan tata letak. Selanjutnya akan dibahas lebih lanjut metode–metode *geo-replication* untuk *cost* efektif yang menggunakan pendekatan tipe data dan tata letak.

#### 1) Tipe Data

Metode replikasi data pada *geo-replication* untuk *cost* efektif dengan menggunakan tipe data terdapat metode yang menggunakan *Conflict Free Replicated Data Type* (CRDT) [23], dan *mergeable data type* [7]. Metode yang menggunakan CRDT merupakan metode *active-active geo replication* untuk membuat sebuah layanan *cache* yang aman dan mudah, serta tidak perlu memerlukan *proxy* atau pihak ketiga jika terjadi konflik data antar server *node*. Metode yang menggunakan *mergeable data type* merupakan metode untuk memudahkan proses

penggabungan pada proses replikasi data pada sisi pemrograman pada fase replikasi. Kedua metode ini memiliki kesamaan yaitu membuat proses replikasi menjadi lebih sederhana dan mudah dilakukan sehingga untuk melakukan proses replikasi tidak perlu memerlukan *cost* atau biaya yang besar, hanya pendekatannya berbeda, salah satunya metode CRDT menghilangkan keterlibatan *proxy* atau pihak ketiga ketika terjadi konflik data, sedangkan *mergeable data type*, memudahkan proses penggabungan beberapa sumber data agar dapat digabungkan dengan aman dan mudah.

## 2) Tata Letak

Metode replikasi data pada *geo-replication* untuk *cost* efektif dengan pendekatan tata letak antara lain, pertama mengajukan algoritma probabilitas untuk menentukan letak server *node* terdekat untuk data yang diinginkan sehingga data dapat diperoleh dengan cepat [16]. Selain itu, terdapat penelitian yang mengusulkan metode LDP, yaitu teknik yang dapat mengefektifkan biaya dengan penempatan pusat data secara efisien khususnya untuk aplikasi sosial media [24]. Cara yang digunakan yaitu dengan menggunakan lokalitas sosial. Aplikasi sosial media berurusan dengan operasi yang sangat interaktif seperti menjelajahi halaman profil beberapa teman atau memposting komentar, yang memerlukan pengambilan data teman. Akan lebih baik jika data teman ditempatkan di lokasi yang sama, karena pengguna tidak perlu pergi ke lebih dari satu pusat data untuk mengambil data temannya, dan menyimpan latensi tambahan dan lalu lintas antar pusat data. Terdapat juga teknik efektifitas biaya khususnya untuk sosial media app dengan menggunakan Graph-Partitioning [25]. Algoritma tersebut bekerja dengan menempatkan data berdasarkan hubungan antar pengguna. Sehingga replikasi data tidak dilakukan di seluruh *data center*.

## B. Latensi Rendah

Metode-metode yang dikembangkan untuk latensi rendah ini antara lain, pertama mengusulkan metode isolasi bernama non-monotonic snapshot isolation untuk transaksi *Atomicity, Consistency, Isolation, Durability* (ACID) [26]. Kemudian ada penelitian yang mengusulkan metode replikasi pada *geo-replication* dengan pendekatan manajemen antara *throughput* dan jarak latensi [27], metode ini memiliki kelebihan yaitu mekanisme yang sederhana dan tetap menjaga konsistensi data yang diperlukan. Penelitian selanjutnya yaitu mengajukan sistem bernama SLOG, yaitu sistem *geo-replication* yang memungkinkan untuk membuat transaksi yang ACID, memiliki *throughput* yang tinggi, dan semua transaksi berjalan dengan latensi rendah [28]. Selain itu, ada penelitian yang mengajukan metode *SwiftCloud*, yaitu metode lain yang dapat memiliki *throughput* tinggi dan latensi rendah [4]. *Swift Cloud* bekerja dengan cara mendistribusikan data antara penyimpanan di sisi klien dan sisi server yang dapat mereplikasi secara efisien. Karena penyimpanan sisi klien yang biasa digunakan di aplikasi web atau mobile dapat meningkatkan waktu respon dan ketersediaan.

Kemudian terdapat penelitian untuk metode GeoPaxos+ yang bekerja secara optimal dalam memberikan latensi yang rendah dengan cara mengelola permintaan *request* menggunakan lokalitas geografis [29]. Cara kerja dari metode ini dengan 2 cara yaitu yang pertama, protokol dapat mengidentifikasi data yang diakses oleh permintaan sebelum permintaan dijalankan. Kedua, protokol dapat menentukan dimana layanan akan diakses. Kemudian pada penelitian lain mengajukan, metode *Droopy* dan *Dripple* yang bekerja dengan dua pendekatan yang mengeksplorasi cara-cara untuk mengurangi resiko masalah *state machine replication* untuk *all leader* dan *leaderless* [6]. Karena pemilihan leader yang tidak tepat dapat mengakibatkan perjalanan bolak-balik yang tidak perlu ke seluruh dunia sehingga mengakibatkan latensi yang tinggi.

Teknik replikasi yang dinamis dirancang untuk mempertahankan lokalitas data semaksimal mungkin dalam mendapatkan kinerja yang tinggi dari aplikasi berskala besar yang didistribusikan secara geografis. Teknik tersebut secara dinamis membuat replikasi data yang sering diakses dekat dengan klien. Skema replikasi yang dinamis memungkinkan penulisan yang memanfaatkan arsitektur dalam sistem penyimpanan terdesentralisasi seperti Dynamo atau Voldemort. Selanjutnya diusulkan algoritma yang memungkinkan klien secara tentatif menemukan replica data terdekat tanpa melakukan permintaan sebelumnya dengan server metadata khusus. Dalam hal ini dilakukan eksperimen implementasi *prototype* dari metode ini memiliki 96 *node* yang didistribusikan di 6 lokasi layanan *cloud* Amazon EC2. Didapatkan hasil eksperimen menunjukkan penurunan latensi hingga 42% dibandingkan dengan solusi canggih berbasis caching lainnya [30].

Kemudian terdapat metode yang membuat latensi rendah dengan algoritma pemilihan urutan penyebaran replika [2]. Pemilihan dilakukan dengan cara memperkirakan latensi setiap penyebaran fungsi evaluasi setiap penyebaran replika berdasarkan *Round-Trip Time* (RTT). Sehingga jika mengurutkan leader replika dan slave yang tepat maka akan mendapatkan latensi tidak tinggi karena Round-trip Time yang rendah. Kemudian pada penelitian lain, protokol replikasi State-machine yang didistribusikan secara geografis memerlukan komunikasi area luas untuk menetapkan permintaan total dari klien. Dalam kasus ini klien tidak tertarik dengan hasil dari permintaan sebenarnya melainkan hanya membutuhkan jaminan bahwa permintaan tersebut akan diproses sehingga biasanya menimbulkan waktu respons atau latency yang terlalu tinggi.

TABEL I  
PENELITIAN GEO REPLICATION YANG BERTUJUAN UNTUK *COST* EFEKTIF DAN LATENSI RENDAH

Makalah	<i>Cost</i> efektif	Latensi rendah	Deskripsi
House et al [23]	V		Metode replikasi dengan menggunakan tipe data CRDT ( <i>Conflict-free Replicated Data Type</i> ) untuk layanan <i>caching</i> data terdistribusi,
Moniz et al [26]		V	Metode replikasi pada sistem <i>Geo-replication</i> bernama <i>blotter</i> untuk transaksi ACID ( <i>Atomicity, Consistency, Isolation, Durability</i> ) dengan latensi rendah.
Matri et al [16]	V		Metode Replikasi data dengan algoritma probabilitas untuk menentukan letak server <i>node</i> terdekat untuk data yang diinginkan, pada penyimpanan data terdistribusi.
Kaki et al [7]	V		Mengusulkan tipe data yang memudahkan untuk pemrograman sistem terdistribusi <i>Geo-replication</i> pada proses replikasi data.
Gunawardhana et al [27]		V	Mengusulkan pendekatan pada sistem <i>Geo-replication</i> yang memperhatikan <i>throughput</i> dan latensi, dengan mekanisme yang sederhana dan tetap menjaga konsistensi data yang diperlukan.
Ren et al [28]		V	sistem <i>geo-replication</i> yang memungkinkan untuk membuat transaksi yang ACID, memiliki <i>throughput</i> yang tinggi, dan semua transaksi berjalan dengan latensi rendah bernama SLOG.
Wu and Madhyastha [21]	V	V	Mengimplementasikan <i>CostLo</i> dan <i>Spanstore</i> untuk <i>web services</i> dengan latensi rendah dan <i>cost</i> efektif.
Mahmood et al [17]	V		Sistem pengimanan <i>Geo-replication</i> bernama <i>Karma</i> , untuk tempat penyimpanan <i>cloud</i> yang <i>cost</i> efektif dan mempertahankan <i>causal consistency</i> .
Liu et al [6]		V	Metode pemilihan leader untuk mengurangi RTT yang tidak perlu sehingga menyebabkan latensi tinggi
Zhou et al [24]	V		Metode pemilihan <i>data center</i> replikasi menggunakan algoritma pemilihan replikasi dengan memanfaatkan lokalitas sosial user.
Coelho and Pedone [29]		V	Algoritma protokol dengan pendekatan lokalitas geografis bernama <i>GeoPaxos+</i> , untuk menentukan <i>node</i> server yang tepat dengan cara mengidentifikasi data yang akan diakses sebelum di proses
Preguiça et al [4]		V	Pendekatan mendistribusikan data antara penyimpanan di sisi klien dan sisi server yang dapat meningkatkan waktu respon dan ketersediaan
Cidon et al [18]	V		Menerapkan replikasi berjenjang pada system penyimpanan <i>cloud</i> untuk mengurangi biaya lebih rendah dalam <i>geo – replication cluster</i> penuh.
Matri et al [30]		V	Metode replikasi data yang dinamis dimana membuat replikasi data yang sering diakses oleh klien menjadi dekat dengan klien, metode ini dapat menurunkan latensi lebih rendah dibandingkan dengan metode <i>caching</i> .
Khalajzadeh et al [25]	V		Metode pemilihan <i>data center</i> replikasi menggunakan algoritma pemilihan replikasi dengan memanfaatkan hubungan antar user..
Yoon et al [22]	V	V	Memberikan tahapan serta evaluasi dalam mencapai latensi dan biaya rendah dalam sistem <i>geo – replication</i> parsial.
Eischer et al [31]		V	Mengusulkan protokol wave dalam replikasi <i>geo</i> berbasis <i>cloud</i> dimana protokol tersebut mengandalkan grup replika berganda dalam wilayah geografis agar klien secara cepat mendapatkan jaminan eksekusi permintaannya.
Ascó [5]	V		Mengusulkan teknik replikasi antar <i>Data center</i> untuk memulihkan jika ada kendala kegagalan situs serta mengurangi biaya akses.
Fouto et al [2]		V	Metode replikasi dengan algoritma pemilihan urutan penyebaran dengan mengevaluasi RTT.
Wu et al [20]	V		Desain metode SPANStore pada beberapa layanan <i>cloud</i> dalam menentukan kebijakan replikasi berdasarkan karakteristik beban kerja dan meminimalkan beban kerja komputasi.



*Weave* adalah solusi yang dihadirkan untuk mengatasi permasalahan tersebut, dalam *wave* protokol replikasi *geo* berbasis *cloud* yang mengandalkan grup replika secara berganda dalam wilayah geografis untuk secara efisien menetapkan nomor urut yang stabil untuk permintaan yang masuk. Pendekatan ini memungkinkan *Weave* untuk menawarkan penulisan terjamin yang memungkinkan klien dengan cepat mendapatkan jaminan eksekusi untuk permintaan mereka [31].

### C. Evaluasi metode – metode replikasi pada *geo-replication*

Pada bagian ini, kami meninjau teknik–teknik dalam *geo-replication* dengan tujuan mencapai *cost* efektif dan latensi rendah. Pada **Error! Reference source not found.** menunjukkan artikel–artikel dimana pembahasannya memecahkan masalah bagaimana cara efisiensi biaya serta mencapai latensi rendah dalam teknik *geo-replication* data di lingkungan terdistribusi.

Artikel ini mensurvei teknik yang canggih dalam *geo-replication* data dimana tujuannya adalah dalam efisiensi biaya dan latensi rendah. Dalam kasus efisiensi biaya kita mengumpulkan metode–metode khusus dari berbagai artikel yang membahas masalah tersebut. Seperti contoh kasus pengoptimalan biaya terhadap aplikasi yang berinteraksi dengan layanan penyimpanan yang didistribusikan secara geografis digunakan teknik yang diberi nama SPANStore. Kemudian teknik replikasi data penuh pada *Data Center* (DC) yang didistribusikan secara geografis dalam meningkatkan ketersediaan data dirasa kurang efisien, diperlukan teknik replikasi data yang dinamis yaitu mereplikasi data hanya di beberapa DC saja sehingga mengurangi biaya. Dalam layanan *cloud* diperkenalkan teknik replikasi tiga arah dalam mengatasi kehilangan data didalam klaster. Teknik tersebut dianggap terlalu berlebihan dan membuat biaya semakin tinggi sehingga diajukan teknik replikasi data berjenjang dimana teknik tersebut membagi klaster menjadi level master dan backup.

Dalam mendapatkan latensi rendah dalam replikasi data dalam lingkungan geografis dalam layanan *cloud* diperlukan teknik replikasi data yang dinamis artinya secara dinamis teknik tersebut membuat replikasi data yang sering di akses dekat dengan klien. Klien secara kondisional menemukan replica data yang terdekat tanpa harus melakukan permintaan ke server metadata khusus. Dengan cara tersebut dapat mengurangi latensi 42% dari metode berbasis *chacing*. Masih didalam lingkungan layanan *cloud*, kita diperkenalkan dengan *protocol wave* dalam replikasi data. *Protocol wave* memberikan solusi mengurangi latensi yaitu dengan memberikan urutan yang stabil dalam permintaan klien sehingga klien secara cepat dalam mendapatkan jaminan eksekusi terhadap permintaan mereka.

Metode replikasi pada *geo-replication* untuk *cost* efektif secara umum yaitu tidak mereplikasi data ke semua *node* server yang ada atau hanya mereplikasi data yang diperlukan atau data yang dianggap penting, dengan kata lain tidak semua data direplikasi. Kedua pendekatan ini yang sering digunakan untuk menekan *cost* atau biaya yang diperlukan untuk proses replikasi data. Kedua pendekatan ini memang berhasil memenuhi kriteria *cost* efektif karena *cost* atau biaya untuk proses replikasi dapat menjadi lebih kecil dibanding dengan proses replikasi secara penuh. Akan tetapi metode replikasi untuk *cost* efektif ini memiliki kelemahan, yaitu ada kemungkinan data yang diperlukan oleh *user* tidak tersedia di semua *node* server, dikarenakan proses replikasi hanya dilakukan ke *node* server tertentu dan atau tidak semua data dilakukan replikasi, sehingga ada kemungkinan data yang diperlukan oleh *user* terdapat pada *node* server yang terletak di lokasi yang jauh dari lokasi *user* yang membutuhkan data tersebut, dan hal ini akan mengakibatkan masalah latensi yang tinggi, karena salah satu yang mempengaruhi latensi adalah jarak antara *node* server dan *user*. Walaupun demikian, terdapat metode replikasi *cost* efektif yang dapat menekan jarak *node* server dan *user* dengan memanfaatkan hubungan antar *user*, sehingga tidak perlu melakukan replikasi ke seluruh DC tanpa mengurangi latensi. Tetapi metode ini hanya bisa diterapkan pada aplikasi yang interaktif antar pengguna seperti sosial media.

Metode replikasi pada *geo-replication* untuk latensi rendah pada umumnya menggunakan beberapa pendekatan yaitu, manajemen *throughput* dan latensi, efisiensi penyebaran replikasi data ke semua *node* server, penentuan *node* server yang terdekat dari lokasi *user* yang membutuhkan, teknik replikasi dinamis untuk menentukan *node* server terdekat dari *user* dan menentukan data yang sering diakses oleh *user* pada umumnya. Manajemen *throughput* dan jarak latensi bertujuan untuk menentukan aliran data yang memiliki *throughput* tinggi sehingga dapat dialokasikan *bandwidth* dan aspek lainnya untuk mengurangi latensi. Efisiensi penyebaran replikasi data ke semua *node* server bertujuan untuk membuat proses replikasi dan penyebaran replikasi data ke semua *node* server bisa dilakukan secepat mungkin agar *user* dapat menggunakan versi replikasi yang terbaru dengan cepat. Pada pendekatan efisiensi penyebaran replikasi data ini terdapat beberapa cara yang digunakan yaitu mengatasi masalah *roundtrip* atau perjalanan bolak balik dari *node* server satu ke *node* server lainnya dan menentukan urutan *node* server dalam mendistribusikan replikasi data. Pendekatan menentukan *node* server yang terdekat dari *user* yang membutuhkan bertujuan untuk mengurangi jarak antara server dan *user* sehingga dapat mengurangi latensi pengiriman data ke *user*. Teknik replikasi dinamis hampir sama dengan menentukan *node* server yang terdekat dari *user* akan tetapi disamping memilih *node* server yang terdekat, teknik replikasi dinamis juga dapat

menentukan data apa yang sering digunakan atau dibutuhkan oleh *user* secara umum, sehingga dapat memprioritaskan data yang sering digunakan atau dibutuhkan ini untuk dilakukan replikasi dahulu daripada data lainnya. Selain itu, terdapat juga teknik pendekatan pendistribusian data dengan memanfaatkan penyimpanan data sementara (*cache*) di sisi klien seperti aplikasi mobile yang dapat meminimalkan latensi dan ketersediaan data. Dari penjelasan beberapa pendekatan untuk metode replikasi pada *geo-replication* untuk latensi rendah ini dapat disimpulkan, metode–metode replikasi pada kategori ini menurunkan latensi dengan mengurangi jarak antara *node* server dan *user*, serta menjaga agar *node* server dalam kondisi *up to date* dengan membuat proses replikasi dilakukan secepat mungkin dari sudut pandang lokasi *user* dan lokasi *node* server.

Pada metode replikasi untuk latensi rendah ini memiliki kelebihan yaitu dapat memberikan kenyamanan untuk *user* yaitu *user* dapat dengan mudah mengakses data yang dia inginkan dengan tanpa perlu menunggu waktu lama. Akan tetapi untuk mengimplementasikan metode replikasi ini diperlukan *cost* atau biaya yang cukup tinggi karena harus melakukan replikasi jika salah satu *node* server terdapat perubahan dan replikasi harus disebarkan ke semua *node* server agar *user* dapat mengakses ke *node* server terdekat dari lokasi dimana dia berada saat ingin mengakses data yang diperlukan. Pada **Error! Reference source not found.** kita dapat mengetahui artikel–artikel mana yang mengusulkan metode replikasi *geo-replication* untuk *cost* efektif dan latensi rendah.

## V. KESIMPULAN

Makalah ini telah membahas tentang metode–metode replikasi data pada sistem *Geo-replication* dengan dari sisi *cost* efektif dan latensi rendah. Dalam makalah ini juga telah didiskusikan tentang pengembangan tipe data dan model tata letak pada metode–metode replikasi dengan pendekatan *Geo-replication*. Metode replikasi untuk *cost efektif* akan cocok digunakan untuk data yang belum bersifat final atau data sementara, dan data transaksi, dimana data–data ini umumnya hanya diperlukan pada *user–user* tertentu saja, tidak semua *user* perlu mengakses data ini. Metode replikasi untuk latensi rendah dapat diimplementasikan untuk data yang sudah final atau data sudah diarsipkan, karena jenis data ini akan digunakan ke berbagai jenis *user* dan diakses untuk pengolahan data selanjutnya seperti sains data.

## DAFTAR PUSTAKA

- [1] T. K. Pandey, I. Singh, dan M. Kumar, “Replication in Distributed Systems and its Improvements,” *Int. J. Curr. Microbiol. Appl. Sci.*, vol. 8, no. 05, hlm. 446–451, Mei 2019, doi: 10.20546/ijemas.2019.805.052.
- [2] P. Fouto, J. Leitão, dan N. Prego, “Practical and Fast Causal Consistent Partial Geo-Replication,” dalam *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, Nov 2018, hlm. 1–10. doi: 10.1109/NCA.2018.8548067.
- [3] AbdullahMSFT, “Active geo-replication - Azure SQL Database.” <https://learn.microsoft.com/en-us/azure/azure-sql/database/active-geo-replication-overview> (diakses 22 Desember 2022).
- [4] N. Prego, M. Zawirski, A. Bieniusa, S. Duarte, V. Balesgas, C. Baquero, dan M. Shapiro, “SwiftCloud: Fault-Tolerant Geo-Replication Integrated all the Way to the Client Machine,” dalam *2014 IEEE 33rd International Symposium on Reliable Distributed Systems Workshops*, Okt 2014, hlm. 30–33. doi: 10.1109/SRDSW.2014.33.
- [5] A. Asco, “Optimising Data Access with an Adaptive Geo-Replication Strategy,” *Int. J. Swarm Intell. Evol. Comput.*, vol. 7, no. 3, hlm. 1–10, 2018, doi: 10.4172/2090-4908.1000171.
- [6] S. Liu dan M. Vukolić, “Leader Set Selection for Low-Latency Geo-Replicated State Machine,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 7, hlm. 1933–1946, Jul 2017, doi: 10.1109/TPDS.2016.2636148.
- [7] G. Kaki, S. Priya, K. Sivaramkrishnan, dan S. Jagannathan, “Mergeable Replicated Data Types,” *Proc ACM Program Lang*, vol. 3, no. OOPSLA, Okt 2019, doi: 10.1145/3360580.
- [8] M. F. Mohamed, “Service replication taxonomy in distributed environments,” *Serv. Oriented Comput. Appl.*, vol. 10, no. 3, hlm. 317–336, Sep 2016, doi: 10.1007/s11761-015-0189-7.
- [9] A. S. M. Noor, N. F. M. Zian, dan F. N. M. S. Bahri, “Survey on replication techniques for distributed system,” *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 2, Art. no. 2, Apr 2019, doi: 10.11591/ijece.v9i2.pp1298-1303.
- [10] B. Alami Milani dan N. Jafari Navimipour, “A comprehensive review of the data replication techniques in the *cloud* environments: Major trends and future directions,” *J. Netw. Comput. Appl.*, vol. 64, hlm. 229–238, Apr 2016, doi: 10.1016/j.jnca.2016.02.005.
- [11] P. Padmanabhan, L. Gruenwald, A. Vallur, dan M. Atiqzaman, “A survey of data replication techniques for mobile ad hoc network databases,” *VLDB J.*, vol. 17, no. 5, hlm. 1143–1164, Agu 2008, doi: 10.1007/s00778-007-0055-0.
- [12] A. Shakarami, M. Ghobaei-Arani, A. Shahidinejad, M. Masdari, dan H. Shakarami, “Data replication schemes in *cloud* computing: a survey,” *Clust. Comput.*, vol. 24, no. 3, hlm. 2545–2579, Sep 2021, doi: 10.1007/s10586-021-03283-7.
- [13] B. A. Milani dan N. J. Navimipour, “A Systematic Literature Review of the Data Replication Techniques in the *Cloud* Environments,” *Big Data Res.*, vol. 10, hlm. 1–7, Des 2017, doi: 10.1016/j.bdr.2017.06.003.
- [14] W. Li, Y. Yang, dan D. Yuan, “A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in *Cloud* Data Centres,” dalam *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, Des 2011, hlm. 496–502. doi: 10.1109/DASC.2011.95.
- [15] “Latency is Everywhere and it Costs You Sales - How to Crush it - High Scalability -.” <http://highscalability.com/latency-everywhere-and-it-costs-you-sales-how-crush-it> (diakses 22 Desember 2022).
- [16] P. Matri, A. Costan, G. Antoniu, J. Montes, dan M. S. Pérez, “Towards Efficient Location and Placement of Dynamic Replicas for Geo-Distributed Data Stores,” dalam *Proceedings of the ACM 7th Workshop on Scientific Cloud Computing*, New York, NY, USA, 2016, hlm. 3–9. doi: 10.1145/2913712.2913715.
- [17] T. Mahmood, S. P. Narayanan, S. Rao, T. N. Vijaykumar, dan M. Thottethodi, “Karma: Cost-Effective Geo-Replicated *Cloud* Storage with Dynamic Enforcement of Causal Consistency,” *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, hlm. 197–211, Jan 2021, doi: 10.1109/TCC.2018.2842184.
- [18] A. Cidon, R. Escriva, S. Katti, M. Rosenblum, dan E. G. Sirer, “Tiered replication: A cost-effective alternative to full cluster geo-replication,” dalam *2015 USENIX Annual Technical Conference (USENIX ATC 15)*, 2015, hlm. 31–43.
- [19] Z. Wu, C. Yu, dan H. V. Madhyastha, “CosTLO: Cost-effective redundancy for lower latency variance on *cloud* storage services,” dalam *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, 2015, hlm. 543–557.
- [20] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, dan H. V. Madhyastha, “Spanstore: Cost-effective geo-replicated storage spanning multiple *cloud* services,” dalam *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, hlm. 292–308.

- [21] Z. Wu dan H. V. Madhyastha, “Cost-Effective Geo-Distributed Storage for Low-Latency Web Services.,” *IEEE Data Eng Bull*, vol. 40, no. 4, hlm. 26–40, 2017.
- [22] H. Yoon, A. Gavrilovska, dan K. Schwan, “Attribute-Based Partial Geo-Replication System,” dalam *2016 IEEE International Conference on Cloud Engineering (IC2E)*, Apr 2016, hlm. 127–130. doi: 10.1109/IC2E.2016.29.
- [23] D. House, H. Kuang, K. Surendran, dan P. Chen, “Toward Fast and Reliable Active-Active Geo-Replication for a Distributed Data Caching Service in the Mobile Cloud,” *Procedia Comput. Sci.*, vol. 191, hlm. 119–126, Jan 2021, doi: 10.1016/j.procs.2021.07.018.
- [24] J. Zhou, J. Fan, J. Jia, B. Cheng, dan Z. Liu, “Location-Aware Data Placement for Geo-Distributed Online Social Networks,” dalam *2016 International Conference on Advanced Cloud and Big Data (CBD)*, Agu 2016, hlm. 234–239. doi: 10.1109/CBD.2016.048.
- [25] H. Khalajzadeh, D. Yuan, J. Grundy, dan Y. Yang, “Cost-Effective Social Network Data Placement and Replication Using Graph-Partitioning,” dalam *2017 IEEE International Conference on Cognitive Computing (ICCC)*, Jun 2017, hlm. 64–71. doi: 10.1109/IEEE.ICCC.2017.16.
- [26] H. Moniz, J. Leitão, R. J. Dias, J. Gehrke, N. Preguiça, dan R. Rodrigues, “Blotter: Low Latency Transactions for Geo-Replicated Storage,” dalam *Proceedings of the 26th International Conference on World Wide Web*, Republic and Canton of Geneva, CHE, 2017, hlm. 263–272. doi: 10.1145/3038912.3052603.
- [27] C. Gunawardhana, M. Bravo, dan L. Rodrigues, “Unobtrusive Deferred Update Stabilization for Efficient  $\{\$Geo-Replication\}$ ,” dalam *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017, hlm. 83–95.
- [28] K. Ren, D. Li, dan D. J. Abadi, “Slog: Serializable, low-latency, geo-replicated transactions,” *Proc. VLDB Endow.*, vol. 12, no. 11, 2019.
- [29] P. Coelho dan F. Pedone, “GeoPaxos+: Practical Geographical State Machine Replication,” dalam *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, Sep 2021, hlm. 233–243. doi: 10.1109/SRDS53918.2021.00031.
- [30] P. Matri, M. S. Pérez, A. Costan, L. Bougé, dan G. Antoniu, “Keeping up with storage: Decentralized, write-enabled dynamic geo-replication,” *Future Gener. Comput. Syst.*, vol. 86, hlm. 1093–1105, Sep 2018, doi: 10.1016/j.future.2017.06.009.
- [31] M. Eischer, B. Straßner, dan T. Distler, “Low-latency geo-replicated state machines with guaranteed writes,” dalam *Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data*, 2020, hlm. 1–9.