

Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data pada Sekolah Musik Indonesia Solo

Willy Sudiarto Raharjo¹, Gani Indriyanta², Amsal Maestro³

Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta

willysr@ti.ukdw.ac.id

ganind@staff.ukdw.ac.id

amsalmaestro@ti.ukdw.ac.id

Diterima 20 Februari 2018

Disetujui 8 Juni 2018

Abstract— Sekolah Musik Indonesia (SMI) Solo is the center of SMI which has internal data using web-based application at appssmi.com site with SQL Server database and has not been backed up regularly. Database replication is a technique for copying and distributing data and database objects from one database to another and implementing synchronization so data consistency can be guaranteed. Replication can be implemented to the cloud by requiring Internet access. The main concern in SMI Solo was the quality access of the Internet connection and also infrastructure used in SMI Solo. The purpose of this research is to analyze the readiness of data replication infrastructure needs at SMI Solo. The results of the analysis are then used as the basis for making recommendations and design of information technology architecture in the implementation of SMI database replication. We concluded that the infrastructure owned by SMI Solo is sufficient to be used for database replication. This is demonstrated by the very satisfactory performance of the SMI Solo server network with 3.85 Mbps download throughput, 3.49 Mbps upload throughput, 0% packet loss, 25.88 ms delay, and 0.09 ms jitter. On database replication performance thorough test scenarios, average performance on snapshot replication is using for CPU 4.78%, DTU 5.94%, I/O 0.06% and log data I/O 5.25%. The average performance on transactional replication is CPU 0.09%, DTU 0.09%, data I/O 0%, and log I/O 0.04%. Some of the challenges in developing database replication infrastructure to be implemented in all SMI's can run efficiently if each SMI has a local server and Internet network albeit with unstable throughput.

Index Terms—Network Performance, Replication, Snapshot, Transactional

I. PENDAHULUAN

Data merupakan aset yang paling berharga dalam dunia digital seperti saat ini. Ketersediaan data menjadi salah satu faktor penting dalam proses bisnis sebuah institusi atau perusahaan karena tanpa adanya data, maka tidak akan ada informasi yang bisa didapatkan untuk pengambilan keputusan atau analisa performa kinerja sebuah perusahaan. Dalam mendukung ketersediaan data di *server*, terdapat

sebuah teknik yang disebut dengan replikasi basis data yang melakukan penyalinan dan pendistribusian data seperti objek-objek basis data dari satu basis data ke basis data lain atau dari media penyimpanan satu ke media penyimpanan yang lain dan melaksanakan sinkronisasi antara basis data sehingga ketersediaan data dapat terjamin.

Dalam melakukan replikasi basis data, tidak mungkin tanpa biaya. Replikasi basis data memerlukan penyimpanan tambahan dan memperbaharui data memerlukan waktu tersendiri [1]. Penyimpanan tambahan yang dimaksud di sisi *server*. Oleh karena itu, *server* memerlukan kesiapan kebutuhannya seperti jaringan Internet yang digunakan untuk mentransfer data. Performa jaringan Internet yang memiliki *bandwidth up to* dapat mempengaruhi keberlangsungan proses replikasi karena *throughput (bandwidth aktual)* selalu berubah-ubah setiap saat.

Sekolah Musik Indonesia (SMI) adalah bidang usaha PT. Sarana Menjangkau Indonesia yang bergerak di dunia pendidikan musik. SMI mempunyai banyak cabang dan karyawan di Indonesia. Sejauh ini, untuk mendapatkan data/informasi mengenai data penting (informasi terkait keuangan, pembelajaran, unit dan investor) yang ada di suatu cabang menggunakan aplikasi berbasis web di situs appssmi.com. Situs appssmi.com menggunakan *database SQL Server* yang di-hosting menggunakan Windows Azure dan belum ada *backup* secara berkala di server milik SMI sendiri. SMI Solo sebagai pusat SMI di Indonesia memiliki Windows Server 2008 R2 Standard yang memiliki 1 IP publik statis dan *bandwidth up to* 4 Mbps.

II. TINJAUAN PUSTAKA

Penelitian tentang replikasi basis data bukanlah hal yang baru, namun pada setiap kasus yang dijumpai, permasalahan yang dicoba untuk diselesaikan memiliki kekhasan sendiri-sendiri.

Silitonga melakukan implementasi replikasi basis

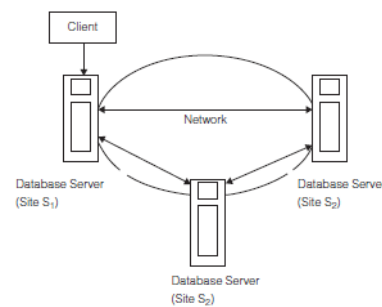
data sistem informasi akademik pada Universitas Katolik Santo Thomas dengan memanfaatkan replikasi multi-master [2], sedangkan Zani dkk mencoba melakukan implementasi metode *asynchronous* pada Replikasi *Database* Inventaris Universitas Bina Darma [3]. Metode *asynchronous* digunakan karena proses replikasi terjadi setelah transaksi di *master* ditahan dulu oleh *buffer* sampai selesai dan dapat diterapkan pada *database open source* seperti MySQL. Mattias juga mencoba melakukan replikasi database multi-master pada layanan E-Learning di Tanzania namun hasil yang di dapatkan ternyata kurang bisa diimplementasikan secara praktis[4].

Truica, Boicea, dan Radulescu telah mencoba melakukan penelitian untuk membandingkan performa dari proses replikasi *asynchronous* basis data Microsoft SQL Server dan PostgreSQL. SQL Server memiliki kelebihan untuk mereplikasikan hanya beberapa bagian pada *database*, kemampuan ini tidak dimiliki oleh PostgreSQL dan MySQL. Untuk performa pengujian replikasi, basis data yang memperoleh waktu yang baik adalah PostgreSQL [5]. Penelitian lain dilakukan oleh Shahapure & Jayarekha untuk membandingkan *performance metric, type*, dan *scalability* pada 11 algoritma replikasi yang dapat diterapkan pada arsitektur *cloud computing*. Hasil penelitiannya adalah setiap algoritma mempengaruhi performa *cloud* ketika melakukan replikasi. Saran dari penelitian ini adalah layanan dengan ketersediaan tinggi memerlukan setidaknya dua server yang identik. Satu *host* utama yang menjalankan layanan, *host* lainnya menjadi tempat cadangan. Dengan menggunakan replikasi, *service* suatu aplikasi dapat ditingkatkan keamanan dan ketersediaan datanya [6].

III. LANDASAN TEORI

A. Replikasi Basis Data

Replikasi merupakan proses menyalin data dari berbagai komputer. Ini merupakan kunci keefektifan dari sistem terdistribusi yang dapat menambah *performance, high availability, dan fault tolerance* [7]. Sebagai contoh, sumber daya dari suatu *web server* di *browser* dan *server web proxy* adalah bentuk replikasi, karena data yang ada di *cache browser* dan pada *server* ada replika dari satu sama lain. Gambaran replikasi ditunjukkan pada Gambar 1.



Gambar 1. Gambaran proses replikasi basis data

Replikasi mempunyai beberapa istilah sebagai berikut:

- *Publisher* : Sebuah *Database Management System (DBMS)* yang membuat data tersedia ke lokasi lain lewat replikasi. *Publisher* bisa satu atau lebih, didefinisikan sebagai set dari objek dan data yang direplikasikan
- *Distributor* : Sebuah DBMS yang menyimpan data replikasi dan metadata tentang publikasi, dalam beberapa kasus bertindak sebagai antrian untuk data bergerak dari *publisher* ke *subscriber*. Sebuah DBMS dapat bertindak sebagai *publisher* dan distributor.
- *Subscriber* : Sebuah DBMS yang menyimpan data replikasi. Subscriber dapat menerima data dari berbagai publisher tergantung kepada tipe replikasi yang dipilih, subscriber dapat melewatkan data perubahan ke penerbit atau menerbitkan data ke *subscriber* lain.

Ada beberapa jenis replikasi yaitu :

- *Snapshot Replication*, suatu *copy* berkala diambil dari *database* master, baik operasi menambah (*insert*) maupun proses memperbaharui (*update*) *database slave*.
- *Synchronous Replication*, data dimasukkan ke dalam kedua *database* master dan *slave* pada waktu yang sama. Jika *database* tidak bisa berkomunikasi, maka kemudian tidak ada data yang dapat dimasukkan, namun bagaimanapun juga *database* selalu disamakan.
- *Asynchronous Replication*, perubahan di dalam master *database* dimasukkan ke suatu antrian dan mentransfer ke *database* lain pada suatu basis periode tertentu. Jika komunikasi atau suatu server *down*, sistem akan menunda perpindahan data.

B. Cloud Computing

Cloud computing adalah sebuah model yang memungkinkan untuk *ubiquitous* (dimanapun dan kapanpun), nyaman, *on-demand* akses jaringan ke sumber daya komputasi yang dapat dengan cepat dirilis atau ditambahkan. *Cloud computing* dapat digunakan dengan berbasis jaringan/Internet [8].

Secara umum arsitektur *cloud computing* terbagi menjadi 3, yaitu [9]:

- *Infrastructure as a Service (IaaS)*. IaaS menyediakan sumber daya pemroses, *storage*, kapasitas jaringan, dan sumber daya komputasi lainnya. Contoh penyedia layanan ini adalah Amazon EC2 dan TelkomCloud.
- *Platform as a Service (PaaS)*. PaaS menyediakan *platform* (bahasa pemrograman, *tools*, web server, basis data) yang berguna untuk pengembangan aplikasi yang berjalan pada infrastruktur *cloud* dan hasilnya dapat dimanfaatkan untuk konsumen. Contoh penyedia layanan ini adalah Azure SQL Database.
- *Software as a Service (SaaS)*. SaaS menyediakan layanan berupa aplikasi yang dapat digunakan oleh konsumen yang berjalan pada infrastruktur *cloud*. Contoh penyedia layanan ini adalah gmail, google docs, Office 365, dan SalesForce.

C. Database Transaction Unit (DTU)

Database Transaction Unit (DTU) adalah satuan/ukuran yang dikembangkan Microsoft untuk membantu pengguna memilih jumlah sumber daya yang dibutuhkan ketika menggunakan Azure SQL Database. Microsoft menggunakan operasi yang unik dalam permintaan *online transaction processing (OLTP)*, lalu mengukur banyaknya transaksi yang dapat diselesaikan per detik di dalam suatu basis data dalam setiap tingkat performa [10].

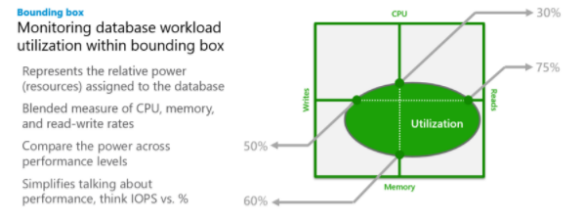
Tabel 1. Pilihan performa DTU

	Basic			Standard				Premium			
	5	10	20	50	100	125	250	500	1,000	1,750	
DTUs	5	10	20	50	100	125	250	500	1,000	1,750	
Max storage (GB)	2	250			500				1,000		
Max in-memory OLTP storage (GB)	N/A	N/A	N/A	N/A	N/A	1	2	4	8	14	
Max concurrent workers	30	60	90	120	200	200	400	800	1,600	2,400	
Max concurrent logins	30	60	90	120	200	200	400	800	1,600	2,400	
Max concurrent sessions	300	600	900	1,200	2,400	2,400	4,800	9,600	19,200	32,000	
Point-in-time restore	Any point last 7 days			Any point last 14 days				Any point last 35 days			
Disaster recovery	Active Geo-Replication, up to 4 offline or online (readable) secondary backups										

Pada Tabel 1, dijelaskan 3 level performa yang bisa dipilih pada Azure SQL Database. Ada level *service* dari DTU yang digunakan. Semakin besar transaksi basis data suatu aplikasi, maka DTU yang diperlukan semakin besar.

Gambar 2 menjelaskan isi dari DTU yaitu *Central Processing Unit (CPU)*, *memory*, transaksi *writes*, dan transaksi *reads*. DTU dapat dipilih menyesuaikan kebutuhan transaksi aplikasi yang menggunakan Azure SQL Database.

Database Transaction Unit – DTU



Gambar 2. Komponen di dalam DTU

IV. METODOLOGI PENELITIAN

A. Pengumpulan Data Survey Lapangan

Proses pengumpulan data dilakukan peneliti melalui kunjungan lingkungan kerja SMI Solo dan melihat secara langsung infrastruktur jaringan yang sudah ada, serta melakukan wawancara kepada tim *software developer* SMI. Kemudian peneliti mengambil data yang diperlukan untuk menganalisis. Data kuantitatif yang akan diambil adalah *throughput*, *packet loss*, *delay*, dan *jitter*. Selain itu peneliti juga mengamati *hardware* yang digunakan sebagai *server* dan mengambil data terkait basis data appssmi.com.

Setelah peneliti mendapatkan data awal, selanjutnya dilakukan analisis terhadap *server*, basis data SQL Server, dan tempat menampung basis data cadangan menggunakan teknologi *cloud computing*. Analisis dilakukan untuk mengetahui apakah kinerja *server*, basis data SQL Server, dan *cloud computing* saling mendukung untuk diterapkan atau tidak.

B. Pengambilan Data Jaringan dan Basis Data

Peneliti menggunakan beberapa alat bantu untuk memperjelas data. Alat bantu yang digunakan untuk mengambil data performa jaringan menggunakan *command prompt* dan *biznet speedtest*. Untuk memetakan struktur basis data peneliti menggunakan aplikasi DBDesigner 4.

C. Analisis Kebutuhan Infrastruktur Replikasi Basis Data

Langkah selanjutnya adalah melakukan analisis kebutuhan infrastruktur replikasi basis data yaitu data jaringan berdasarkan standarisasi TIPHON [11]. Pengolahan data pengamatan disajikan dalam bentuk grafik dan tabel sehingga mudah untuk menarik kesimpulan dan mengidentifikasi masalah. Hasil pengujian dilihat dari data yang dikumpulkan berupa data *bandwidth test (throughput)*, *delay*, *packet loss*, dan *jitter* yang disajikan dalam bentuk tabel dan grafik. Hasil evaluasi tersebut menjadi dasar untuk penerapan replikasi basis data yang akan diimplementasikan. Tahapan ini diharapkan sudah dapat menjadi jaringan yang siap digunakan pada lingkungan kerja SMI Solo menerapkan replikasi basis data.

D. Perancangan dan Implementasi Replikasi Basis Data

Peneliti merancang dan mengimplementasikan replikasi basis data sesuai dengan pertimbangan kepada *developer* SMI Solo. Pada tahap ini, peneliti mengimplementasikan replikasi menggunakan teknologi Windows Azure yaitu Azure SQL Database. Ada beberapa batasan yang dilakukan ketika melakukan replikasi dari *server* lokal ke Azure SQL Database sehingga menjadi bahan pertimbangan penerapan replikasi.

E. Monitoring

Setelah replikasi sudah diuji coba dan berhasil, tahap berikutnya adalah menganalisis beberapa tantangan yang ada sesuai dengan kesiapan infrastrukturnya SMI Solo kedepannya. Terakhir, menarik kesimpulan dari segala proses yang telah dikerjakan.

V. HASIL DAN PEMBAHASAN

A. Pembuatan Alat Uji Jaringan

Alat ukur yang digunakan untuk mengukur kualitas dari jaringan Internet *server* adalah *ping test* dan *Biznet Speedtest*. *Ping* merupakan *program tool* dari sistem operasi Windows melalui *command prompt* yang berfungsi untuk mengetahui jumlah *delay* dan *packet loss*. *Biznet Speedtest* adalah aplikasi *online* yang digunakan untuk mengetahui berapa kecepatan koneksi (*throughput*) yang sedang dipakai oleh *user*. Hasil dari pemantauan menggunakan *ping* dan *speedtest* diolah menjadi grafik dan dihitung rata-ratanya. Pengujian ini dilakukan pada *server* SMI Solo.

B. Hasil Pengujian Performa Jaringan

Peneliti ingin mengetahui kesiapan kebutuhan replikasi basis data pada SMI Solo dapat dilakukan atau tidak dengan kondisi saat ini. *Server* yang digunakan pada saat ini tidak yakin digunakan oleh *developer* karena bukan Internet dengan *bandwidth* yang *dedicated* sehingga dengan pengujian jaringan ini dapat meyakinkan *developer* kedepannya.

Pengambilan data Internet *server* SMI Solo dilakukan pada jam kerja selama 12 sesi (setiap interval 30 menit) pada hari Senin sampai Sabtu pukul 12.00 – 18.00 WIB. Data yang diambil yaitu *throughput*, *packet loss*, *delay*, dan *jitter* sesuai dengan standarisasi QoS [12]. Data kemudian diolah menjadi data statistik.

Untuk standarisasi hasil pengujian *throughput*, *delay*, *jitter*, dan *packet loss* digunakan acuan dari TIPHON seperti pada tabel 2, 3, 4.

Tabel 2. Standarisasi *throughput* TIPHON

Category	Throughput
Very Good	100%

Good	75%
Average	50%
Bad	< 25%

Tabel 3. Standarisasi *delay* TIPHON

Category	Delay
Very Good	< 150ms
Good	150 – 300 ms
Average	300 – 450 ms
Bad	> 450 ms

Tabel 4. Standarisasi *jitter* TIPHON

Category	Delay
Very Good	0 ms
Good	0 – 75 ms
Average	75 – 125 ms
Bad	125 – 225 ms

Tabel 5. Standarisasi *packet loss* TIPHON

Category	Packet Loss
Very Good	0%
Good	3%
Average	15%
Bad	25%

C. Rekapitulasi dan Analisis Data Performa Jaringan

1) Analisis *Throughput* Internet

Setelah data *throughput* didapatkan, kemudian diolah dengan menggunakan Microsoft Excel dan ditampilkan dengan perbandingan rata-rata dalam bentuk Tabel 6. Secara penilaian TIPHON, rata-rata nilai menunjukkan bahwa pada *throughput download* dan *upload* termasuk dalam kategori sangat memuaskan.

Tabel 6. Rekapitulasi pengukuran *throughput server* SMI Solo

Hari	Throughput Download (Mbps)	Throughput Upload (Mbps)	Limitasi (Mbps)	Rata-rata Pengguna Harian	Standarisasi TIPHON
Senin	3,78	3,36	4	1-10	Sangat Bagus
Selasa	3,87	3,48			Sangat Bagus
Rabu	3,94	3,47			Sangat Bagus
Kamis	3,55	3,49			Sangat Bagus
Jumat	3,97	3,6			Sangat Bagus
Sabtu	4	3,54			Sangat Bagus
Rata-rata	3,85	3,49	4	1-10	Sangat Bagus

Rata-rata *throughput download* sebesar 3,85 Mbps dan *throughput upload* sebesar 3,49 Mbps dengan pengguna harian 1-10 pengguna. Dapat dikatakan pula pemanfaatan *throughput* Internet *server* jika akan jadi *server* lokal untuk domain *appssmi.com* menunjukkan performa jaringan yang baik.

2) Analisis *Packet Loss*

Pengujian *packet loss* dengan menghitung manual paket yang hilang/loss dari *command prompt* saat tes *ping* menuju www.google.com. Pengujian dilakukan sebanyak 72 kali dengan setiap pengujiannya melakukan permintaan *echo* sebanyak 10 *byte*.

Berdasarkan data yang diambil peneliti, dapat disimpulkan bahwa nilai *packet loss* Internet *server* SMI Solo sangat rendah karena tidak ada paket yang hilang dengan nilai 0% (kategori penilaian TIPHON sangat bagus).

3) Analisis *Delay*

Tabel 7 menjelaskan *delay server* SMI Solo. *Delay* tertinggi terdapat pada hari Selasa dengan nilai 29,19 ms. Sedangkan *delay* terendah berada di hari Sabtu dengan nilai 22,45 ms. Secara keseluruhan nilai rata-rata *delay* untuk Internet *server* SMI Solo memiliki nilai 25,88 ms dimana dalam standarisasi TIPHON termasuk kategori sangat bagus.

Tabel 7. Rekapitulasi hasil pengukuran *delay server* SMI Solo

Hari	Delay (ms)	Standarisasi TIPHON
Senin	23,31	Sangat Bagus
Selasa	29,19	Sangat Bagus
Rabu	26,59	Sangat Bagus
Kamis	28,65	Sangat Bagus
Jumat	25,42	Sangat Bagus
Sabtu	22,45	Sangat Bagus
Rata-rata	25,88	Sangat Bagus

4) Analisis *Jitter*

Tabel 8 menjelaskan performa *jitter* di *server* SMI Solo. *Jitter* tertinggi terdapat di hari Rabu dengan nilai 0,19 ms dan terendah di hari Kamis dan Jumat dengan nilai 0,05 ms. Secara keseluruhan nilai rata-rata *jitter* sebesar 0,09 dan termasuk ke dalam kategori bagus berdasarkan penilaian TIPHON.

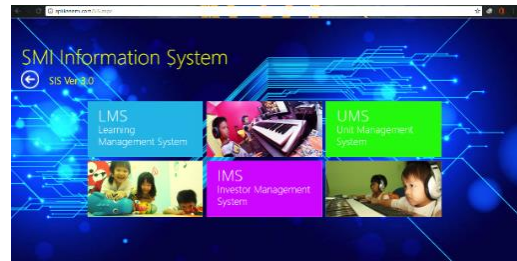
Tabel 8. Rekapitulasi hasil pengukuran *jitter server* SMI Solo

Hari	Jitter (ms)	Standarisasi TIPHON
Senin	0,07	Bagus
Selasa	0,1	Bagus
Rabu	0,19	Bagus
Kamis	0,05	Bagus
Jumat	0,05	Bagus
Sabtu	0,06	Bagus
Rata-rata	0,09	Bagus

D. Perancangan dan Implementasi Replikasi Basis Data

1) Membuat domain *aplikasismi.com*

Pada Gambar 3, peneliti membuat domain lain dengan basis data *appssmi.com* untuk kepentingan replikasi sehingga mencegah hal-hal yang tidak diinginkan. Domain ini datanya di-hosting pada *server* lokal SMI Solo.



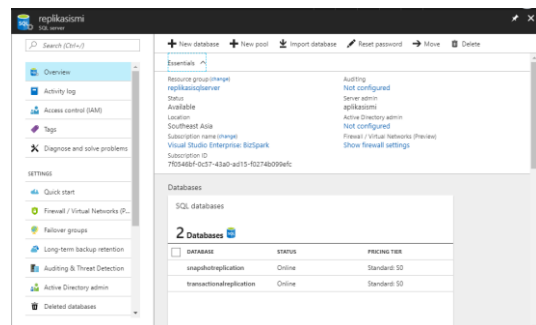
Gambar 3. Tampilan situs *aplikasismi.com*

2) *Server* lokal SMI Solo sebagai tempat hosting

Basis data dan data *website* *aplikasismi.com* ditempatkan pada *server* lokal SMI Solo. *Server* lokal dipilih untuk menjadi tempat utama *website* dikarenakan spesifikasi *server* dan performa jaringan yang sangat memuaskan.

Dengan perancangan seperti ini, peneliti akan melakukan replikasi ke Azure SQL Database. Azure dipilih karena teknologi *cloud* yang aman, selalu *online* 24 jam, rekomendasi developer *website* SMI, dan memiliki *interface* yang sederhana sehingga mudah dimengerti.

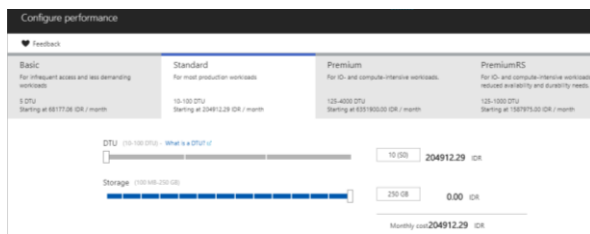
3) Membuat *instance Server* dan Azure SQL Database di Microsoft Azure



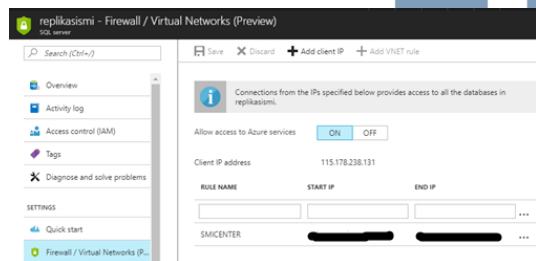
Gambar 4. Membuat 1 *instance server* dan 2 Azure SQL Databases di Microsoft Azure

Gambar 4 menjelaskan ketika peneliti ingin membuat 1 atau lebih Azure SQL Database, memerlukan setidaknya 1 *server* untuk menampung *database* tersebut. Peneliti menentukan lokasi di Southeast Asia dengan pertimbangan dekat, murah, dan memiliki *latency* yang paling rendah dari Indonesia.

Setiap Azure SQL Database memiliki *performance* yang berbeda-beda disesuaikan dengan kebutuhan pengguna. Peneliti memilih *performance* dengan tipe *standard* dengan minimal 10 DTU dan *storage* sebesar 5GB dijelaskan pada Gambar 5. Peneliti menggunakan 10 DTU dikarenakan sudah pernah mencoba menggunakan tipe *basic* dan hasilnya ada beberapa proses DTU akan *overload* hingga 100% sehingga akan mengurangi proses kecepatan replikasi basis data. Untuk ukuran *database* bisa memilih antara 1 – 250 GB tanpa dikenakan biaya, peneliti hanya memilih muatan *database* sebesar 5GB.



Gambar 5. Pilihan *performance* tipe *Standard* untuk Azure SQL Database



Gambar 6. Menambahkan pengaturan *firewall*

Gambar 6 menjelaskan ketika *server* akan diakses suatu *client*, maka IP *client* tersebut harus dikonfigurasi pada pengaturan *firewall*. Peneliti mengkonfigurasi *rule name*, *start IP*, dan *end IP* yang merupakan identitas *server* SMI Solo. *Start IP* dan *end IP* sengaja dirahasiakan karena IP publik statis *server* SMI Solo.

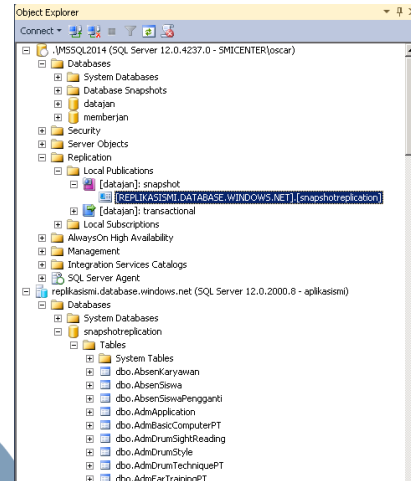
4) Proses Replikasi menggunakan SQL Server Management Studio 2014

Gambar 7 menjelaskan ketika konfigurasi replikasi sudah berhasil dilakukan, maka akan ada keterangan tambahan yang muncul pada saat selesai konfigurasi *Publisher* dan *Subscriber*. Peneliti membuat jadwal replikasi *snapshot* pada waktu terpendek yaitu selalu *update* selama 1 jam sekali (pengaturan waktu *backup* dapat disesuaikan dengan kebutuhan, peneliti melampirkan konfigurasi di lampiran B). Jadi, setiap 1 jam akan ada proses besar untuk replikasi jenis ini.

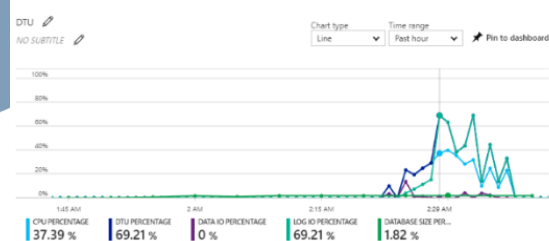
Pada Gambar 8, digambarkan beberapa grafik terhadap proses terjadinya replikasi jenis *snapshot* pertama kali. Terlihat DTU dan Log I/O yang paling tinggi prosesnya sebanyak 69,21%, CPU bekerja

sebanyak 37,39% dan *database size* terisi 1,82% dari total 5GB.

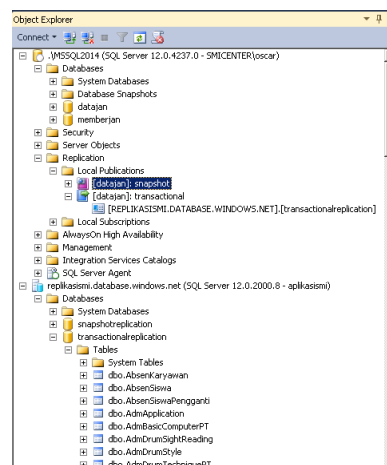
Gambar 9 menjelaskan ketika konfigurasi replikasi sudah berhasil dilakukan, maka akan ada keterangan tambahan yang muncul pada saat selesai konfigurasi *Publisher* dan *Subscriber*.



Gambar 7. Tampilan *snapshot replication* di SSMS 2014

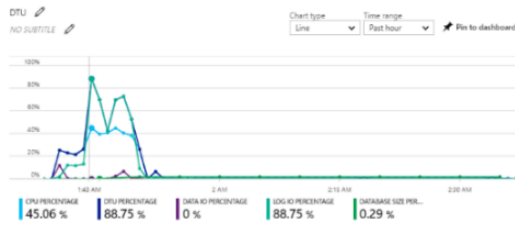


Gambar 8. Proses replikasi pertama kali pada jenis *snapshot replication*



Gambar 9. Tampilan *transactional replication* di SSMS 2014

Pada Gambar 10, digambarkan beberapa grafik terhadap proses terjadinya replikasi jenis *transactional* pertama kali. Terlihat DTU dan Log I/O yang paling tinggi prosesnya sebanyak 88,75%, CPU bekerja sebanyak 45,06% dan *database size* baru terisi 0,29% dari total 5GB.



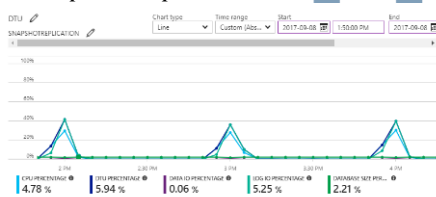
Gambar 10. Proses replikasi pertama kali pada jenis *transactional replication*

E. Hasil Pengujian Replikasi Basis Data

Pengujian replikasi basis data menggunakan 4 skenario yang di-submit pada waktu yang sama dengan diulangi selama 5 kali dengan 13 klien yang berperan sebagai *customer service*. Data yang dihasilkan kemudian akan diolah menjadi data statistik pada Microsoft Excel. Setelah nilai hasil replikasi telah didapat, lalu dibandingkan dengan yang lainnya.

Besarnya DTU yang terpakai dipengaruhi dari Log Input/Output dan Central Processing Unit yang terpakai. Berikut perbedaannya melalui DTU yang digunakan (lebih jelas jika dilihat lewat *transactional replication*) :

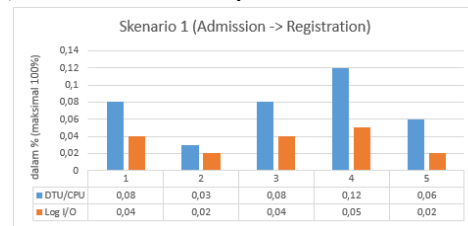
1) Snapshot Replication



Gambar 11. Rata-rata proses *snapshot replication* selama pengujian

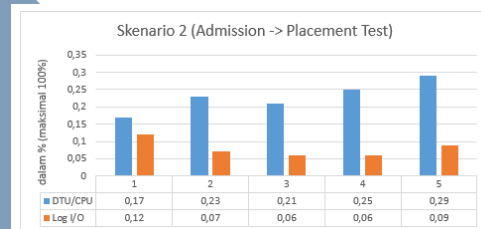
Gambar 11 menjelaskan rata-rata *performance* dari CPU, DTU, data I/O, log I/O, dan *database size* pada *snapshot replication* selama pengujian. Rata-rata *performancenya* 4,78% pada CPU, 5,94% pada DTU, 0,06% pada data I/O, 5,25% pada log I/O, dan ukuran *database* sebesar 2,21% dari 5GB. *Database* yang dikonfigurasi replikasi jenis *snapshot* akan besar penggunaan DTUnya setiap 1 jam sekali (sistem otomatis melakukan replikasi selama 20 menit). Penggunaan DTU tergolong cukup besar setiap 1 jam sekali karena sistem akan meng-overwrite data yang ada.

2) Transactional Replication



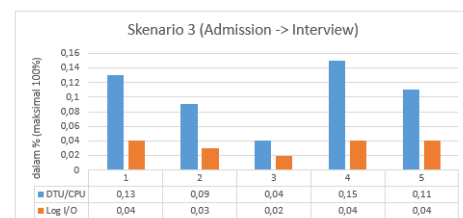
Gambar 12. Grafik *performance* hasil replikasi ke Azure untuk Skenario 1

Pada Gambar 12 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O pada skenario 1. Pada proses replikasi jenis transaksional, DTU dan CPU bernilai sama. Untuk proses DTU yang paling banyak ada di percobaan keempat sebanyak 0,12% dan paling sedikit di percobaan kedua sebanyak 0,03%. Input paling banyak terjadi pada percobaan keempat sebanyak 0,05% dan paling sedikit di percobaan kedua dan terakhir sebanyak 0,02%.



Gambar 12. Grafik *performance* hasil replikasi ke Azure untuk Skenario 2

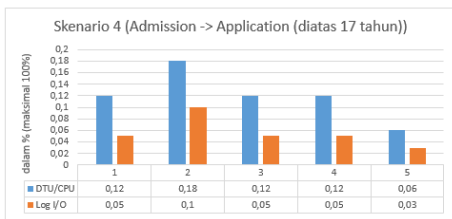
Pada Gambar 13 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 2. Untuk proses DTU yang paling banyak ada di percobaan kelima sebanyak 0,29% dan paling sedikit di percobaan pertama sebanyak 0,17%. Input paling banyak terjadi pada percobaan pertama sebanyak 0,12% dan paling sedikit di percobaan ketiga dan keempat sebanyak 0,06%.



Gambar 14. Grafik *performance* hasil replikasi ke Azure untuk Skenario 3

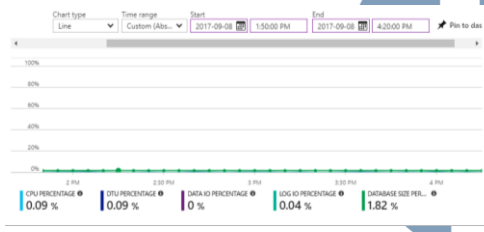
Pada Gambar 14 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 3. Untuk proses DTU yang paling banyak ada di percobaan keempat sebanyak 0,15% dan paling sedikit di percobaan ketiga sebanyak 0,17%. Input paling banyak terjadi pada percobaan pertama, keempat, dan

kelima sebanyak 0,04% dan paling sedikit di percobaan ketiga sebanyak 0,02%.



Gambar 15. Grafik *performance* hasil replikasi ke Azure untuk Skenario 4

Pada Gambar 15 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 4. Untuk proses DTU yang paling banyak ada di percobaan kedua sebanyak 0,18% dan paling sedikit di percobaan kelima sebanyak 0,06%. Input paling banyak terjadi pada percobaan kedua sebanyak 0,1% dan paling sedikit di percobaan kelima sebanyak 0,03%.



Gambar 16. Rata-rata proses *transactional replication*

Gambar 16 menjelaskan rata-rata *performance* dari CPU, DTU, data I/O, log I/O dan *database size* dari *transactional replication* selama pengujian. Rata-rata *performancenya* 0,09% pada CPU, 0,09% pada DTU, 0% pada data I/O, 0,04% pada log I/O, dan ukuran *database* sebesar 1,82% dari 5GB. Penggunaan DTU tergolong kecil karena setiap waktu sistem akan menyalin data setiap kali berubah.

F. Analisis Replikasi Basis Data

1) Snapshot Replication

Pengujian *snapshot replication* dapat dilihat secara detail hanya beberapa saat ada proses-proses ringan yang terjadi tetapi tidak mengubah isi *database* jika belum saatnya sinkronisasi otomatis yang sudah dikonfigurasi. Pengujian selama proses itu berlangsung, data tidak akan langsung ter-*update* sehingga harus menunggu sinkronisasi berikutnya.

Pada replikasi jenis *snapshot*, rata-rata dari 10 DTU yang dialokasikan, dipakai sebanyak 40%-an selama pengujian. Proses ini terbilang memakan banyak *resources* dan tidak efisien karena *database* akan selalu di-*overwrite* datanya selama waktu yang ditentukan.

2) Transactional Replication

Tabel 9. Rekapitulasi hasil pengukuran *performance transactional replication*

Skenario	DTU/CPU (%)	Log I/O (%)	Utilisasi (%)
1	0,074	0,034	100
2	0,23	0,08	
3	0,104	0,034	
4	0,12	0,056	
Rata-rata	0,132	0,064	

Setelah data *performance* didapatkan, kemudian diolah dengan menggunakan Microsoft Excel dan ditampilkan dengan perbandingan rata-rata dalam pada Tabel 9. Melalui rata-rata tersebut terlihat penggunaan DTU/CPU besar pada skenario 2 dan kecil pada skenario 1. Sedangkan untuk log I/O besar pada skenario 2 dan kecil pada skenario 1 dan 3.

Secara pengamatan, didapati bahwa dengan 4 skenario yang tergolong berat (berdasarkan pengamatan peneliti dan developer appssmi) dengan 13 klien *submit* bersamaan, replikasi jenis transaksional prosesnya sangat ringan dengan 10 DTU. Oleh karena itu, jika nanti Azure SQL Database di-*downgrade* performanya menjadi 5 DTU dapat menghemat biaya dan performa tetap terjamin berjalan dengan baik.

G. Analisis Tantangan Pengembangan Infrastruktur Replikasi Basis Data SMI

1) Analisis Perkembangan Basis Data Appssmi.com

Data internal SMI dengan domain appssmi.com per 22 Oktober 2017 memiliki basis data yang berukuran 97 MB di Azure (Azure memiliki algoritma kompresi data, ukuran data sebenarnya bila di-*download* dari Azure bermuatan 310 MB). Data ini digunakan oleh 12 cabang SMI di Indonesia dan cabang selalu bertambah setiap tahunnya.

Tabel 10. *Traffic* basis data appssmi.com selama 12 hari

Tanggal	Jumlah Data Masuk (KB)	Jumlah Transaksi
9 Oktober 2017	380,69	404
10 Oktober 2017	498,63	531
11 Oktober 2017	355,58	378
12 Oktober 2017	387,18	413
13 Oktober 2017	258,09	277
14 Oktober 2017	220,09	237
16 Oktober 2017	362,25	388
17 Oktober 2017	283,36	295
18 Oktober 2017	391,6	418
19 Oktober 2017	222,41	247
20 Oktober 2017	300,34	321
21 Oktober 2017	258,75	278
Rata-rata per hari	326,58	348,92

Dilihat dari perkembangan data selama 12 hari pada Tabel 10, maka peneliti memiliki estimasi jika 1

tahun appssmi.com (sementara dengan 12 cabang) yang buka selama 317 hari (sudah dikurangi libur hari minggu setahun ada 48, tanpa libur nasional) dikali rata-rata data masuk perhari yaitu 326,58 KB akan menambah data sebanyak 103525,86 KB (103,53 MB dalam desimal) secara linier. Muatan ini tergolong kecil untuk 12 cabang karena isinya teks.

2) Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data di Setiap Cabang SMI

Dengan data yang akan selalu bertambah, kesiapan kebutuhan infrastruktur dan performa jaringan Internet di setiap cabang SMI perlu ditingkatkan dan diefisiensikan untuk menambah ketersediaan data yang terdistribusi. Di masa yang akan datang, peneliti menganalisis terkait infrastruktur secara khusus dalam pengelolaan jaringan Internet yang baik untuk mengimplementasikan replikasi basis data.

Jaringan Internet di setiap cabang SMI merupakan kebutuhan primer. Hal ini didasari oleh setiap proses belajar mengajar dan pengelolaan data internal memerlukan Internet. Jaringan Internet dengan *bandwidth dedicated* harganya lebih mahal bila dibandingkan dengan jaringan Internet dengan *bandwidth up to* (karena tingkat kestabilan dan kehandalan).



Gambar 17. Pengembangan arsitektur teknologi SMI di masa depan

Dari Gambar 17, dijelaskan setiap cabang memiliki *server* lokal sendiri yang menampung data internal perusahaan. Dengan adanya dukungan *server* lokal, koneksi untuk mengakses data internal SMI akan cepat dan stabil (tidak terlalu tergantung Internet) sehingga penggunaan Internet akan terfokus terhadap proses belajar mengajar selama jam operasional SMI. Selanjutnya, bila jam SMI tutup, masing-masing *server* sudah diberikan jadwal (setiap *server* cabang untuk melakukan replikasi tidak boleh sama jadwalnya 1 dengan yang *server* cabang lainnya) untuk melakukan *merge replication* agar setiap cabang melakukan proses *backup* basis datanya ke *server* pusat terlebih dahulu sampai semua cabang selesai. Akhirnya, *server* pusat memiliki jadwal untuk melakukan *merge replication* sekali lagi ke setiap *server* cabang agar semua mendapat basis data yang sama.

VI. KESIMPULAN

Setelah melakukan serangkaian penelitian, pengujian, dan analisa, maka kesimpulan dari penelitian ini adalah sebagai berikut :

- Kesiapan kebutuhan infrastruktur replikasi basis data SMI Solo sudah baik dengan *bandwidth* Internet server SMI Solo dengan *up to* 4 Mbps dan basis data internal yang kecil (hanya teks).
- Kinerja dari jaringan *server* SMI Solo memiliki kualitas yang sangat memuaskan dilihat dari rata-rata *bandwidth download* 3,85 Mbps, *bandwidth upload* 3,49 Mbps, *packet loss* 0%, *delay* 25,88 ms, dan *jitter* 0,09 ms.
- Kinerja dari replikasi basis data dalam hal penggunaan *resource* CPU, DTU, data I/O, log I/O, dan *database size* dapat dilihat dari :
 - Rata-rata penggunaan CPU dari *snapshot replication* yaitu 4,78% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,09%.
 - Rata-rata penggunaan DTU dari *snapshot replication* yaitu 5,94% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,09%.
 - Rata-rata data I/O dari *snapshot replication* yaitu 0,06% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0%.
 - Rata-rata log I/O dari *snapshot replication* yaitu 5,25% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,04%.
 - Rata-rata *database size* dari total 5GB pada *snapshot replication* yaitu 2,21% dan *transactional replication* yaitu 1,82%. Perbedaan ini didasarkan pada data yang direplikasikan pada proses *transactional replication* semua tabel yang direplikasikan harus memiliki *primary key*, sedangkan tabel "trx" pada appssmi.com tidak memiliki *primary key* sehingga tidak dapat direplikasikan.

Saran untuk penelitian selanjutnya adalah mencoba membandingkan dengan layanan cloud computing lain sebagai perbandingan dengan apa yang dimiliki oleh Azure, seperti halnya Google Cloud Computing dan Alibaba Cloud Computing. Amazon Web Service tidak disertakan karena sampai dengan saat tulisan ini dibuat, AWS belum mendukung fitur replikasi yang dibutuhkan. Selain itu juga bisa melakukan perbandingan algoritma replikasi data seperti pada [13] dan [14].

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Sekolah Musik Indonesia Solo beserta tim pengembang appssmi.com yang telah mendukung penelitian ini dengan menyediakan infrastruktur serta akses pada arsitektur sistem yang digunakan.

DAFTAR PUSTAKA

- [1] M. Christian Mazilu, "Database Replication", Database Systems Journal Vol 1, No 2, hal 33-38, 2010.
- [2] P. Silitonga, "Replikasi Basis Data Pada Sistem Pengolahan Data Akademik Universitas Katolik Santo Thomas". Jurnal TIME Vol III No 1. hal. 32-36, 2014.
- [3] F. Zaini, L. Atika, & U. Ependi, "Implementasi Metode Asynchronous pada Replikasi Database Inventaris Universitas Bina Darma", Jurnal Informatika Universitas Bina Darma Palembang, Oktober 2015.
- [4] M. Holmgren, "Multi-Master Database Replication and e-Learning – Theoretical and Practical Evaluation" (Disertasi), 2015.
- [5] C-O. Truica, A. Boicea, & F. Radulescu, "Asynchronous Replication in Microsoft SQL Server, PostgreSQL, and MySQL". International Conference on Cyber Science and Engineering Hal. 50-55, 2013.
- [6] N.H. Shahapure & P. Jayarekha, "Replication: A Technique for Scalability in Cloud Computing", International Journal of Computer Applications Vol 122, No 5, hal 13-18, July 2015.
- [7] G. Coulouris, J. Dollimore, T. Kindberg, & G. Blair, "Disributed Systems: Concepts and Design, 5th Ed", Pearson, 2011.
- [8] P. Mell & T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, September 2011.
- [9] A. Syaikh, "Komputasi Awan (Cloud Computing) Perpustakaan Pertanian", Jurnal Pustakawan Indonesia Vol 10, No 1, hal 1-12, 2010.
- [10] C. Rabeler, "Migrating SQL Server Databases to Azure, 1st Ed", 2016.
- [11] The European Telecommunications Standards Institute, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON): General aspects of Quality of Service (QoS)", ETSI TR 101 329, France, Vol 2.1.1, June 1999.
- [12] M. Flanagan, R. Froom, & K. Turek, "Cisco Catalyst QoS: Quality of Service in Campus Networks 1st Ed", 2003.
- [13] MC. Lee, FY. Leu, YP. Chen, "PFRF: An adaptive data replication algorithm based on star-topology data grids", Future Generations Computer Systems, Vol 28 Issue 7, July 2012.
- [14] JW. Lin, CH. Chen, J.M. Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems", IEEE Transactions on Cloud Computing, Vol 1 Issue 1, June 2013.