

# METODE PENJAMINAN KUALITAS PADA TEKNOLOGI CLOUD COMPUTING

Sinung Suakanto <sup>#1</sup>

Departemen Teknik Elektro – Institut Teknologi Harapan Bangsa

Jl. Dipati Ukur 80-84 Bandung

Telp. (022) 2506636

<sup>1</sup> sinung@ithb.ac.id

**Abstrak**— Saat ini cloud computing dikenal sebagai teknologi baru yang mengembangkan teknologi grid dengan menerapkan teknik virtualisasi sebagai sumber untuk melakukan komputasi. Teknologi ini mempunyai filosofi bahwa sumber komputasi, penyimpanan dan lain-lain tidak perlu lagi dilakukan di masing-masing client. Akan tetapi semuanya dapat dikerjakan di suatu server yang berada di jaringan internet (cloud). Untuk mendukung teknologi ini diperlukan sebuah jaringan akses internet yang berkecepatan tinggi untuk mendukung transmisi data dengan cepat. Teknologi ini mempunyai kendala performansi sehubungan dengan performansi jaringan akses yang digunakan serta performansi di sisi server sendiri. Metode penjaminan kualitas diperlukan untuk memberikan jaminan bahwa setiap pengguna dapat menggunakan aplikasi atau layanan cloud computing sesuai dengan kesepakatan layanan atau service level agreement. Paper ini berfokus pada review beberapa penelitian yang telah membangun metode penjaminan kualitas pada teknologi cloud computing atau sejenis. Dari beberapa review paper tersebut maka paper ini juga mengusulkan arsitektur metode penjaminan kualitas pada teknologi cloud computing.

**Kata Kunci**— penjaminan kualitas, cloud computing

**Abstract**— Nowadays, Cloud computing known as new technology that develop grid technology with virtualization as computation resource. It has philosophy that computation resource, storage, etc would not compute at each client but compute at server on internet cloud. To realize this technology, it needs internet access with broadband bandwidth and high speeds network for faster data transmissions. Implementation of this technology limited on network access bandwidth and also server performance itself. Quality of Service (QoS) guarantee method was needed to realize service that responding to the service level agreement (SLA). This paper focused at reviewed some research finding at developing Quality of Service (QoS) on cloud computing or similar method. At the end, we also propose new approach for Quality of Service guarantee architecture on cloud computing.

**Keywords**— Quality of Service (QoS) guarantee, cloud computing

## I. PENDAHULUAN

Teknologi internet saat ini telah berkembang pesat yang menuju kepada generasi *distributed computing* dan *cloud computing*. *Cloud computing* atau komputasi awan mempunyai filosofi dimana sumber komputasi, sumber data akan ditempatkan di suatu server yang berada di internet (*cloud*). Dengan kondisi seperti ini maka bakal diprediksi

bahwa trafik di jaringan akan tumbuh sangat pesat dan ada kemungkinan akan berkali lipat jika kecenderungannya mengarah kepada situasi dimana perusahaan besar atau lembaga pemerintah menerapkan teknologi *cloud* ini. Hal yang diandalkan dari teknologi adalah komunikasi akses antara client dan server (*cloud server*). Eksploitasi dan eksplorasi terhadap jaringan ini menjadi isu penting *cloud computing* karena dengan teknologi jaringan pita lebar sudah bisa diwujudkan. Akan tetapi, jika eksploitasi terhadap jaringan ini tidak diiringi dengan penjaminan kualitas yang baik maka akan menyebabkan era komputasi awan susah terwujud atau terimplementasi dengan baik. Penjaminan kualitas pada layanan aplikasi di internet dan komputasi awan saat ini dan masa yang akan datang akan menjadi sebuah isu yang penting.

Penelitian tentang mekanisme penjaminan kualitas pada cloud computing memang masih minim. Beberapa penelitian lebih mengarah ke proyek pengembangan *Quality of Service* (QoS) seperti TAPAS, RESERVOIR dan QoS-Aware Cloud [7][8][14]. Permasalahan penjaminan kualitas adalah adanya SLA (*Service Level Agreement*) antara pengguna dengan *service provider* yang belum dijamin ketika diimplementasikan di jaringan publik. Hal ini terkait dengan delay dari jaringan publik yang secara administratif di luar kendali dari *service provider*. Permasalahan lain muncul juga ketika pengguna dengan profil premium dengan biaya paling mahal, tetapi ketika dia berada pada posisi geografis yang jauh dari *cloud server* maka delay tidak dapat dihindarkan. Dalam hal ini kepuasan pengguna akan turun terhadap *service provider* sebagai penyedia layanan. Permasalahan ini merupakan akibat dari belum adanya atau belum bagusnya mekanisme penjaminan kualitas pada cloud yang mengkaitkan dengan kategori profil pengguna cloud sesuai dengan SLA (*Service Level Agreement*) antara pengguna dengan *service provider*.

Implementasi layanan cloud saat ini banyak menggunakan protokol yang umum digunakan untuk komunikasi akses antara client dan server yang kebanyakan masih menggunakan protokol HTTP. Komunikasi kepada pusat-pusat server cloud tersebut dapat dilakukan menggunakan HTTP GET, PUT and DELETE commands via HTTP atau sering disebut sebagai web service REST [6]. Problem yang ada adalah layanan berbasis web (HTTP) termasuk termasuk *web service* seperti RESTfull merupakan layanan yang karakteristik alaminya

adalah "best effort service". Layanan ini tidak memberikan jaminan kualitas sehingga setiap pengguna akan saling berebut sumber daya (bandwidth, dan sebagainya). Sehingga terdapat teori umum dari kejadian empiris bahwa jika jumlah trafik yang mengakses jaringan cukup tinggi maka performansi layanan cenderung melambat.

Adapun metode-metode penjaminan kualitas yang sudah banyak dikembangkan saat ini kebanyakan lebih ditujukan untuk aplikasi real-time seperti IP-multimedia, Voice over IP (VoIP) dan sebagainya. Metode penjaminan kualitas yang lain seperti pada MPLS (*Multiprotocol Label Switching*) diimplementasikan dengan melakukan perubahan terhadap devais jaringan atau konfigurasi jaringan. Router yang ada harus diubah dengan router yang mendukung MPLS. Hal ini cukup menyulitkan untuk implementasi penjaminan kualitas pada jaringan yang sudah ada yang secara administratif dimiliki secara publik.

## II. CLOUD COMPUTING

### A. Pendahuluan

Cloud computing merupakan sebuah paradigma pragmatis penyediaan layanan internet yang merupakan perkembangan dari teknologi parallel dan grid computing. Cloud computing sendiri merupakan kombinasi dari perkembangan teknologi web (web 2.0, REST, dll), sistem komputasi tersebar (*grid computing*), dan kemampuan untuk manajemen mandiri (*autonomic computing*) [11]. Jika sebelumnya kecenderungan penggunaan layanan dilakukan secara skala kecil maka cloud computing memberikan paradigma baru bahwa proses dan komputasi dapat dijalankan atau diakses dari luar. Akses data dapat dilakukan dari mana saja menggunakan perangkat *fixed* atau *mobile device* menggunakan *internet cloud* sebagai sumber komputasi atau tempat menyimpan data atau fungsi-fungsi yang lain [12].

Cloud computing sendiri sebagai implementasi dari konsep *on-demand computing* model dan *utility computing*. *On-demand computing* model merupakan model yang mengijinkan service provider untuk memberikan komputasi resource tersedia kepada pengguna ketika diperlukan saja. [7]. Adapun *Utility Computing* mempunyai tujuan untuk maksimasi efisiensi penggunaan resource dan minimasi biaya enterprise [7].

Dalam sudut pandang cloud, semua sumber daya dan kapabilitas dari cloud diberikan kepada pengguna sebagai sebuah service lewat internet tanpa teknik atau pengetahuan tertentu atau bahkan mengontrol infrastruktur yang mengendalikannya [11]. Cloud computing sendiri diawali dari perkembangan teknologi sebelumnya seperti *grid computing* yang memungkinkan berbagai, memilih atau agregasi dari sumber daya yang tersebar secara geografis termasuk superkomputer, sistem penyimpanan (storage) atau perangkat yang dimiliki organisasi yang berbeda [12]. Selain itu pada *grid computing* juga sudah dikenal konsep *clustering*. *Clustering* merupakan arsitektur yang dapat digunakan untuk meningkatkan baik *throughput* maupun *availability* dari aplikasi terdistribusi dengan teknik *load balancing* dan *replication*.

### B. Arsitektur Umum Cloud Computing

Arsitektur Cloud Computing adalah cukup bervariasi tergantung dari kebutuhan untuk penyelenggaraan layanan. Beberapa teknologi open source seperti Eucalyptus telah memberikan banyak gambaran tentang arsitektur umum dari teknologi cloud computing. Arsitektur ini secara umum dapat dilihat seperti pada gambar 1. Secara umum terdapat beberapa komponen utama dalam arsitektur sebuah cloud computing berdasarkan Eucalyptus [3] :

- **Node Controller (NC)**

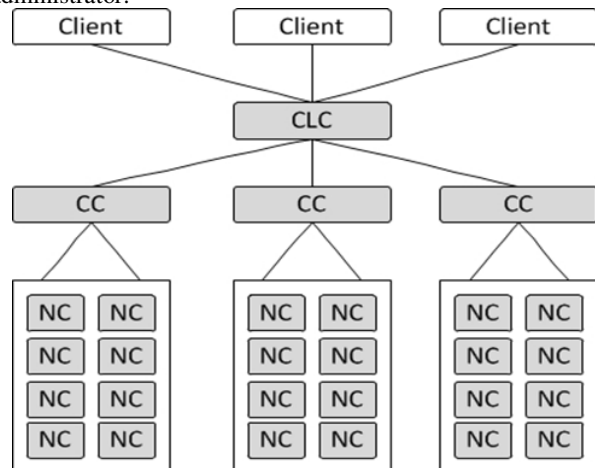
Merupakan komponen yang mengeksekusi sumber daya fisik yang menjalankan Virtual Machines (VM) termasuk bertanggung jawab untuk start up, inspeksi, shutdown serta cleanup.

- **Cluster Controller (CC)**

CC mempunyai fungsi untuk mengumpulkan informasi dari kumpulan NC yang dibawahinya, termasuk di dalamnya menjadwalkan eksekusi VM ke NC tertentu, manajemen konfigurasi jaringan private atau publik. Sekumpulan dari NC yang secara logik bersama-sama melapor ke Cluster Controller yang biasanya dijalankan pada cluster head node atau server yang mengakses jaringan baik private maupun publik.

- **Cloud Controller (CLC)**

CLC mempunyai fungsi untuk memproses permintaan yang data dari user atau permintaan administrator, membuat penjadwalan VM secara high-level, memproses Service Level Agreement (SLA) dan mengadminstrasikan informasi sistem dan pengguna. Setiap Eucalyptus sistem memiliki CLC tunggal yang merupakan gerbang masuk pengguna maupun administrator.



Gambar 1. Arsitektur Cloud Computing pada Eucalyptus [3]

## III. PENJAMINAN KUALITAS PADA CLOUD COMPUTING

### A. Penjaminan Kualitas pada Cloud Computing

Kualitas Layanan (*Quality of Service/QoS*) pada *Cloud Computing* merupakan nilai kualitas dari sebuah layanan *Cloud Computing* yang diselenggarakan oleh service provider dan diterima oleh pengguna. Berbeda dengan layanan web pada jaringan *best-effort* dimana tidak terdapat perjanjian

dengan pengguna maka pada konteks Cloud Computing maka terdapat perjanjian antara service provider dengan pengguna yang tertuang dalam bentuk SLA (*Service Level Agreement*). Sehingga manajemen QoS pada konteks Cloud Computing merupakan usaha atau teknik untuk menelaraskan kualitas layanan operasional dengan dengan SLA. SLA merupakan kontrak atau perjanjian yang menyatakan jaminan QoS yang dibutuhkan oleh para pengguna. Secara umum meliputi [8]:

- Parameter teknis seperti maksimum *response time*, *throughput*, *error rate*.
- *Availability* (tingkat ketersediaan atau tingkat berapa persen down).
- Parameter non teknis seperti : Skalabilitas

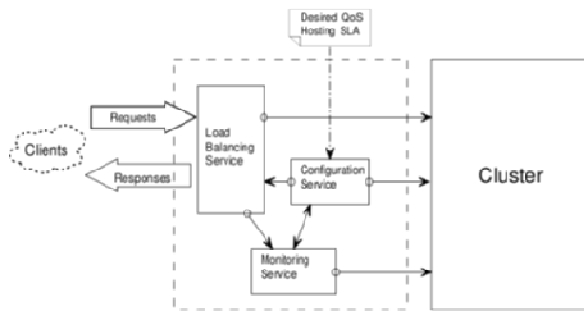
Beberapa model QoS yang telah dikembangkan untuk Cloud Computing di antaranya adalah :

- *Trusted and QoS-Aware Provision of Application Services* (TAPAS) [7]
- *REsources and SERvices Virtualization withOut BarrieRs* (RESERVOIR) [8][14]
- *Qos-Aware Cloud* [8]

### B. TAPAS

Arsitektur TAPAS dapat digambarkan seperti pada gambar 2 di bawah ini. Terdapat dua middleware service yang berperan penting dalam sistem penjaminan kualitas yaitu :

- *Configuration Services*  
Bertanggung jawab untuk mencatat konfigurasi termasuk nilai Service Level Agreement yang telah ditetapkan untuk masing-masing pengguna.
- *Monitoring Service*  
Befungsi untuk memonitorng terjadinya penyimpangan terhadap QoS yang seharusnya diterima. Meminta kepada Configuration Services untuk mengubah nilai SLA jika terjadi penyimpangan dari SLA



Gambar 2. Arsitektur QoS pada Cloud Computing menggunakan TAPAS

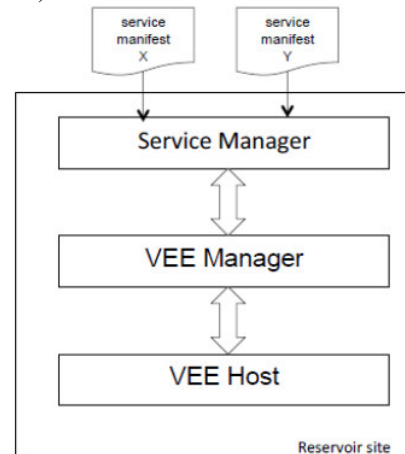
### C. RESERVOIR

Arsitektur RESERVOIR dapat dilihat seperti pada gambar 2.4. RESERVOIR mempunyai 3 level arsitektur yaitu :

- *VEE Host*  
Befungsi untuk memonitor dan mengendalikan *Virtual Machine* (VM).
- *VEE Manager*  
Memanajemen dan mengatur terhadap beberapa VEE Host

### ▪ Service Manger

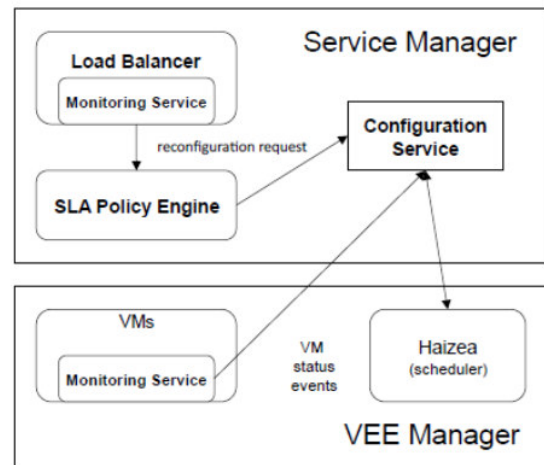
Pemilihan aplikasi berdasarkan service manifest (SLA)



Gambar 3. Arsitektur QoS pada Cloud Computing menggunakan RESERVOIR

### D. QoS-Aware Cloud

*QoS-Aware Cloud* merupakan penggabungan dari TAPAS dengan RESERVOIR. Salah satu perubahannya adalah mengembangkan bagian Service Manager menjadi seperti TAPAS. Arsitekturnya dapat dilihat seperti pada gambar 4 di bawah ini.



Gambar 4. Arsitektur QoS pada Cloud Computing menggunakan QoS-Aware Cloud

Konsep ini merupakan pengembangan dari metode SLA pada QoS-aware Cluster yang dikembangkan oleh Lodi [7]. SLA pada metode ini dicatat dalam bentuk XML file yang terdiri dari Client Responsibilities dan Server Responsibilities yang mencerminkan hak dan kewajiban aplikasi client dan aplikasi server. Aturan di client harus menentukan jumlah maksimum request yang diijikkan untuk mengirimkan aplikasi dalam selang waktu tertentu (*requestRate*). Contoh *requestRate*="100/s".

Aturan di server contohnya adalah sebagai berikut :

```

serviceAvailability="0.99"
efficiency="0.95"
efficiencyValidity="2"
maxResponseTime="1.0s

```

*ServiceAvailability* menentukan probabilitas dimana aplikasi yang dihosting harus available selama periode waktu tertentu. Persentase dari pelanggaran SLA yang dapat ditoleransi dalam selang waktu yang telah ditetapkan, sebelum service provider aplikasi mendapatkan penalty (ekonomi). Contoh diatas menyatakan bahwa tidak kurang dari 95% dari permintaan pengguna dapat dilayani dalam selang waktu 2 jam. Adapun *maxResponseTime* merupakan nilai efektif untuk mengukur tingkat *responsive* dari sebuah service.

Tujuan utama dari arsitektur yang dikembangkan oleh Lodi adalah :

1. Menjamin kualitas sesuai yang telah ditentukan dalam SLA
2. Optimasi penggunaan resource sehubungan dengan tujuan nomor satu
3. Maksimasi portabilitas dari software arsitektur

Arsitektur yang dikembangkan juga mempunyai fitur seperti kemampuan adaptasi menggunakan *feedback control*. Selain itu, karena sistem bekerja pada lingkungan cluster yang berkarater variabilitasnya tinggi dan kondisi beban yang tidak terprediksi, mekanisme *load balancing* akan dibutuhkan. Mekanisme ini akan membantu untuk menyeimbangkan beban permintaan pengguna berdasarkan beban aktual sehingga mencegah sebuah server terhadap kelebihan beban.

Proses konfigurasi cluster terdiri dari membangun cluster awal dengan membentuk sekumpulan nodes dari seminimal mungkin node yang ada untuk menjamin layanan sesuai dengan SLA. Adapun proses konfigurasi runtime terdiri atas proses yang secara dinamis mengubah ukuran cluster dengan menambah atau mengurangi node sesuai yang dibutuhkan. Penambahan node dibutuhkan untuk menangani pertumbuhan beban dan pada saat kasus dimana node dari cluster mengalami kegagalan fungsi dan perlu digantikan perannya. Sedangkan pengurangan node dilakukan dalam rangka untuk mengoptimalkan penggunaan sumber daya.

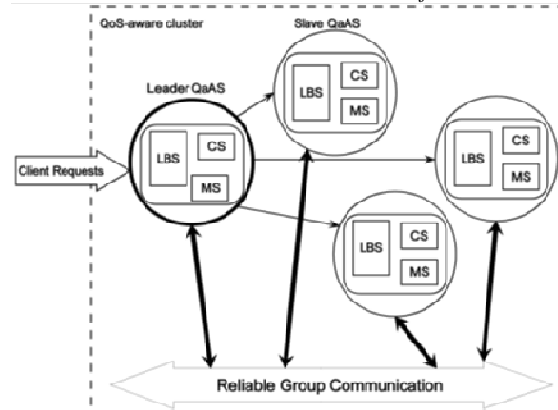
Komponen *monitoring service* mempunyai fungsi memonitor kemungkinan : variasi dari keanggotaan cluster, variasi dari performansi cluster dan pelanggaran terhadap SLA.

Adapun komponen *Load balancing* mempunyai fungsi seperti :

1. Menangkap setiap permintaan HTTP dari client
2. Memilih node target yang dapat melayani permintaan dengan menggunakan beberapa aturan load balancing
3. Meneruskan permintaan client ke node yang terpilih
4. Menerima balasan dari node yang terpilih
5. Meneruskan balasan kepada client yang mengirimkannya.

Cara kerja dari sistem ini dapat dilihat seperti pada gambar 5. Node QoAS yang pertama kali terpakai akan menjadi *Leader*. *Configuration Services* pada *Leader* akan memparsing SLA input untuk diekstrak menjadi parameter QoS yang digunakan untuk menentukan konfigurasi cluster yang diperlukan (Diantaranya adalah client *requestRate*,

*serviceAvailability*, *efficiency*). *Monitoring Service* bertanggung jawab untuk mengecek performansi cluster. Untuk tujuan ini, *Monitoring Service* terdiri dari tiga komponen yaitu *SLA Violations Monitors*, *Evaluation and Violation Detection Service* dan *Cluster Performance Monitor*.



Gambar 5. Arsitektur QoS pada QoS-aware cluster

Untuk setiap permintaan HTTP, dimana response time dari cluster masih memenuhi SLA yang ditetapkan serta memicu untuk rekonfigurasi cluster jika diperlukan. Keputusan untuk memicu rekonfigurasi cluster berdasarkan pada nilai batas peringatan dan sekumpulan nilai indicator dari Cluster Performance Monitor. Pada penelitian tersebut tidak semua indicator digunakan tetapi hanya akan menggunakan index efisiensi saja.

Index efisiensi memungkinkan Monitoring Service untuk mengecek trend kecepatan pelanggaran oleh cluster. Index ini dimulai dari angka 0 hingga turun ke nilai integer negatif dimana nilai tersebut dapat di-increment dan di-decrement.

Jika  $\mathcal{E}$  merupakan efisiensi SLA. Index efisiensi didecrement

dengan  $\left(\frac{1}{1-\mathcal{E}}\right)-1$  yang merupakan nilai aktual response time permintaan yang melanggar maksimum Response Time SLA. Nilai tersebut akan di-increment (ditambah 1) jika tidak terjadi pelanggaran SLA hingga mencapai 0.

Sebagai contoh seperti yang dicontohkan pada bagian sebelumnya bahwa SLA menyebutkan bahwa nilai response time boleh dilanggar hingga maksimum 5% ( $\mathcal{E} = 95\%$ ). Kemudian jika terdapat setiap permintaan client, jika cluster tidak memenuhi SLA response time untuk sebuah permintaan maka indeks efisiensi di-decrement hingga

$\left(\frac{1}{1-0,95}\right)-1 = 19$ . Jika sebaliknya maka akan di-increment sebanyak 1. Karakteristik ini akan menjaga index diantara (-19,0). Sebaliknya jika dibawah -19 maka kemungkinan besar akan terjadi pelanggaran SLA.

#### IV. CLOUD COMPUTING

Adapun penelitian yang terkait dengan penelitian pada QoS dapat diperbandingkan hasilnya seperti pada tabel 3.2 di bawah ini

TABEL 3.2 PENELITIAN QoS PADA CLOUD COMPUTING

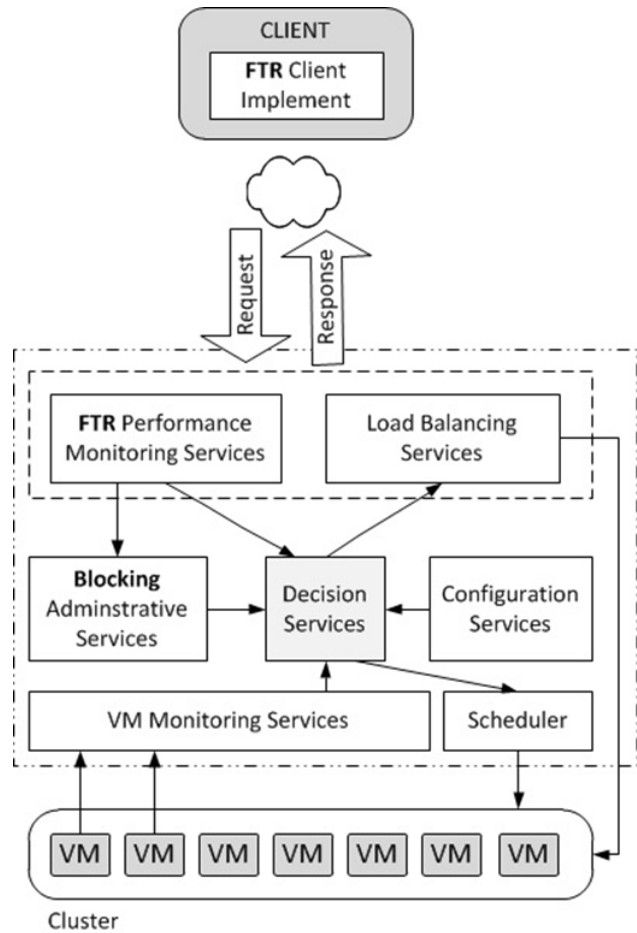
Metode	Deskripsi Metode	Fokus Pengembangan	Model QoS yang dikembangkan	Metode yang digunakan
1) QoS MOS (2010)	Adaptive Service-Based System	Mentranslasikan kebutuhan QoS high-level menjadi formulasi probabilistic temporal, yang mempunyai kemampuan adaptasi dengan menganalisa konfigurasi sistem yang optimal.	Mengembangkan service-based system untuk mencapai kebutuhan QoS dengan kemampuan adaptif untuk menyesuaikan terhadap kondisi sistem, lingkungan serta beban	Pemecahan Analitis menggunakan Markov Model (DTMC dan MDP).  Solusi eksak berdasarkan model probabilistic checking menggunakan eksperimen PRISM
Multiple QoS Constrained Scheduling Strategy of Multi-Workflows (MQMW) (2009) [20]	Service-Based System pada Cloud Computing dengan multi-workflow	Fokus pada algoritma penjadwalan workflow yang berawal dari kapan saja.	Mempertimbangkan banyak parameter QoS pada algoritmanya.  Mempertimbangkan QoS yang berbeda untuk pengguna yang berbeda (multi class)	Hybrid Heuristic for DAG Scheduling
RESERVOIR	Arsitektur Middleware untuk Penjaminan QoS	Membangun arsitektur QoS dengan konsep layering untuk mempermudah dalam pengembangannya	Arsitektur QoS dengan konsep Layering	Implementasi Penjaminan QoS Menggunakan J2EE
QoS-Aware Cluster/ QoS-aware Application Servers (QaASs) (2007)	Arsitektur Middleware pada topologi Cluster	Distributed enterprise applications dengan topologi Cluster	Penjaminan Kualitas sesuai dengan SLA	Menggunakan parameter seperti serviceAvailability, efficiency, efficiencyValidity untuk menjamin kualitas atau maksimum pelanggaran terhadap SLA
QoS-Aware Cloud (2010)	Arsitektur Middleware pada Cloud Computing	Menjamin kualitas sesuai yang telah ditentukan dalam SLA. Optimasi penggunaan resource sehubungan dengan tujuan nomor satu	Menggabungkan arsitektur RESERVOIR dengan TAPAS	J2EE based technology QoS-aware
Trust Evaluation Model for QoS Guarantee in Cloud Systems (2010)	Estimasi availability untuk setiap resource pada periode tertentu, sehingga cloud bisa mengkonfigurasi system secara dinamis dan mendistribusi tugas secara efisien	Model untuk rekonfigurasi dan alokasi dinamis secara efisien untuk memenuhi berbagai kebutuhan pengguna	Alokasi resource computing yang berkualitas tinggi pada pengguna	Probabilistic Latent Semantic Analysis (pLSA) untuk mengestimasi availability untuk setiap service dari data historis

## V. USULAN ARSITEKTUR PENJAMINAN KUALITAS PADA CLOUD COMPUTING

Beberapa penelitian-penelitian tersebut diatas menunjukkan beberapa hal yang perlu diamati lebih lanjut :

- Arsitektur hanya berfokus pada monitoring dan pencatatan performansi QoS dan belum menekankan bagaimana teknik penjaminan kualitasnya
- Manajemen kualitas masih berfokus pada manajemen kualitas server internal atau terkait dengan cluster dan belum mempertimbangkan bagaimana mengatasi problem delay yang disebabkan oleh jaringan
- Beberapa paper lebih berfokus pada pengembangan aplikasi atau middleware (software development) dibandingkan berfokus kepada pengembangan metode penjaminan kualitas dan analisis analitiknya.

Untuk itu penelitian ini mengajukan model arsitektur untuk pembangunan QoS pada cloud computing seperti yang terlihat pada gambar 6. Arsitektur yang dikembangkan berdasarkan pada riset sebelumnya tentang mekanisme penjaminan kualitas menggunakan teknik blocking pada FTR-HTTP [19]. Prinsip sederhana dari teknik ini adalah bagaimana cara memberikan jaminan kepada paket data yang masuk dengan cara mengurangi paket-paket data yang berasal dari pengguna dengan performansi yang kurang baik. Dengan teknik ini diharapkan dapat menjaga kualitas beberapa user yang memang masih berada pada level performansi yang baik. Teknik FTR-HTTP ini dikombinasikan dengan sebuah framework penjaminan kualitas pada server cloud computing.



Gambar 6. Arsitektur Implementasi

Aristektur tersebut mempunyai beberapa komponen seperti :

1. *Configuration Services*  
Komponen ini berfungsi untuk mencatat profile SLA untuk setiap masing-masing profile pengguna.
2. *Monitoring Services*  
Komponen ini berfungsi untuk memonitoring performansi dari eksternal dan internet. Terdapat 2 kategori yaitu :
  - a. Eksternal *Monitoring Services* : Untuk memonitor dan mencatat performansi dari masing-masing user. Di antaranya untuk mencatat history performansi dari client yang dikerjakan oleh *FTR-Performance Monitoring Services*. Sedangkan untuk blocking dari client dicatat oleh *Blocking Administrative Services*.
  - b. Internal *Monitoring Services* : Untuk memonitor dan mencatat performansi internal yang berfokus pada performansi dari *Virtual Machine (VM)*.
3. *Decision Services*  
Komponen ini berfungsi sebagai sistem pengambil keputusan yang secara otomatis jika melihat

penuruna atau kekurangan performansi maka aksi-aksi apa yang perlu dikerjakan untuk memperbaiki performansi tersebut.

## VI. KESIMPULAN

Paper ini telah membahas berbagai metode penjaminan kualitas pada cloud computing dengan berbagai contoh dan teknik implementasinya. Paper ini ingin menunjukkan bahwa riset di bidang ini belum terlalu banyak. Paper ini juga mengusulkan sebuah framework umum untuk mekanisme penjaminan kualitas pada cloud computing berdasarkan apa yang telah dikembangkan pada penelitian sebelumnya. Penelitian selanjutnya akan dilanjutkan untuk membangun simulasi dan skenario implementasi serta bagaimana framework ini diuji.

## REFERENSI

- [1] Radu Celinescu, "Dynamic QoS Management and Optimisation in Service-Based Systems", IEEE Transactions on Software Engineering, 2010
- [2] Kuan-Ta Chen, Polly Huang, Chin-Luang Lei, "Effect of Network Quality on Player Departure Behaviour in Online Games", IEEE Transaction on Paraleler and Distributed Systems, vol.20, No.5, May 2009
- [3] Daniel Nurmi et al, "Eucalyptus : A Tehenical Report on an Elastic Utility Computing Architeccture Linking Your Programs to Useful Systems"
- [4] Daniel Nurmi et al, "Eucalyptus : The Eucalyptus Open-source Cloud-computing System"
- [5] Eddy Caron, Frederic Desprez, David Loureiro, "Cloud Computing Resource Management through a Grid Middleware: A Case Study with DIET and Eucalyptus", 2009 IEEE International Conference on Cloud Computing
- [6] Simson L. Garfinkel, "An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS"
- [7] G. Lodi, F. Panziera, D. Rossi and E. Turrini, "SLA-Driven Clustering of QoS-Aware Application Servers", IEEE Trans. on Soft. Eng. 33(3), pp.186-197, 2007
- [8] Fabio Panziera, Michele Pellegrini, Elisa Turrini, "QoS-Aware Clouds"
- [9] Mohammad Alrifai and Thomas Risse, "Efficient QoS-aware Service Composition"
- [10] Johnson D et al, "Eucalyptus Beginner's Guide", 2010
- [11] Zhenyu Liu, "Agent-based Online Quality Measurement Approach in Cloud Computing Environment", 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology
- [12] V. D. Cunsolo, S. Distefano, A. Puliafito, M. Scarpa, "Applying software engineering principles for designing Cloud@Home", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing
- [13] Rajkumar Buyyaa, Chee Shin Yea, Srikumar Venugopala, James Broberg, Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", ELSEVIER Future Generation Computer Systems, 2008
- [14] Reservoir project. <http://www.reservoir-fp7.eu/>
- [15] Chia-Feng Lin, Ruey-Shyang Wu, Shyan-Ming Yuan, Ching-Tsorng Tsai, "A Web Services Status Monitoring Technology for Distributed System Management in the Cloud", 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery
- [16] Hyukho Kim, Hana Lee, Woongsup Kim, Yangwoo Kim, "A Trust Evaluation Model for QoS Guarantee in Cloud Systems", International Journal of Grid and Distributed Computing, Vol.3, No.1, March, 2010
- [17] Giorgia Lodi, Fabio Panziera, Davide Rossi, Elisa Turrini, "QoS-aware Application Server: Design, Implementation, and Experimental Evaluation"
- [18] Tnigo Goiri, Ferran Juli'a, J. Oriol Fit'o, Mario Mac'ias, and Jordi Guitart, "Resource-Level QoS Metric for CPU-Based Guarantees in Cloud Providers"
- [19] Sinung Suakanto, Suhono Harso Supangkat, Suhardi, "Introduction to Finite Time Response-HTTP: a Simplest Way to Guarantee Quality of Service of Web Application under Best Effort Network", Proceedings International Conference AOTULE 2010