SISTEM PENDETEKSI INTRUSI BERBASIS *MOBILE AGENT* PADA KOMPUTASI AWAN (*CLOUD COMPUTING*)

Fahmi Dzikrullah[1)], Selo Sulistyo [2)], Noor Akhmad Setiawan [3)]

1), 2), 3) Jurusan Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada, Yogyakarta
Jl. Grafika 2, Kampus UGM, Yogyakarta 55281
Email : fahmy.dzik.mti13@mail.ugm.ac.id [1)], selo@ugm.ac.id [2)], noorwewe@ugm.ac.id [3)]

Abstrak

Dewasa ini, cloud computing semakin populer di industri teknologi informasi, akan tetapi kebanyakan pengguna *cloud tidak tahu menahu bagaimana proses pertukaran data dalam jaringan cloud, padahal cloud sangat rentan terhadap ancaman keamanan dan pihak provider seringkali menyembunyikan kelemahan itu, oleh karena itu keamanan pada cloud menjadi salah satu masalah yang perlu diperhatikan sebagai salah satu strategi "pencegahan lebih penting dari pada pengobatan", dengan mendeteksi potensi celah keamanan sedini mungkin lebih baik daripada merespon sebuah serangan setelah sistem sudah dalam keadaan bahaya. Dalam paper ini akan membandingkan beberapa model pendeteksian penyusup pada sebuah lingkungan komputasi cloud terdistribusi, dan menjelaskan beberapa metode pencegahan dari sistem pendeteksi penyusup atau dikenal sebagai intrusion detection system (IDS) dan mengusulkan model solusi IDS berbasis mobile agent yang di aplikasikan pada cloud client dan di harapkan akan menagani beberapa kekurangan IDS pada cloud computing.*

Kata kunci: *Cloud computing, Intrusion detection system, Mobile agent.*

1. Pendahuluan

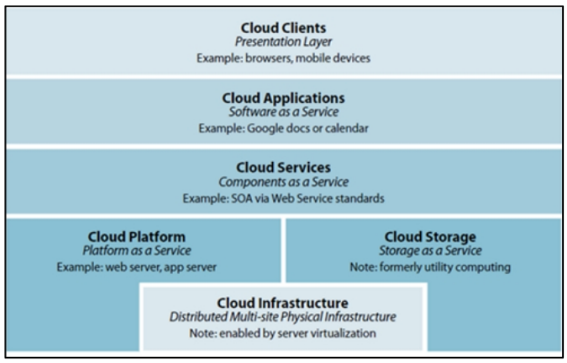
Menurut Onno W Purbo, "*Cloud computing* adalah sebuah model komputasi dimana sumber daya seperti *computing processor, storage, network* dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan atau internet mengunakan pola akses *remote*" atau dengan kata lain memindahkan layanan penyimpan data dan komputasi lokal yang digunakan sehari-hari ke Internet, dimana *virtual shared server* menyediakan *software, infrastruktur, platform, devices* dan sumber daya yang lainnya sebagai layanan pada pengguna, pengguna *cloud* tidak membeli infrastruktur fisik akan tetapi mereka menyewa pada pihak *provider cloud*, yang diperlukan pengguna hanya koneksi internet, pengguna mengunakan sumber daya sebagai suatu layanan (*as a service*) baik sebagai *private cloud* atau *public cloud* tanpa harus mengetahui dan terlibat dalam bagaimana mengelola sumber daya *cloud*. Flexibelitas, kecepatan, kemudahan menjadikan teknologi *cloud yang* bermanfaat di dunia industri teknologi informasi, yang memungkinkan kegiatan sehari-hari untuk dilakukan secara singkat.

Berdasarkan data tahun 2013 pengguna layanan *cloud* di dunia diperkirakan mencapai 500 juta, tumbuh 66,67% dari tahun sebelumnya yang sebanyak 300 juta.[2]. namun dibelakang semua *euforia cloud* itu kebanyakan pengguna tidak tahu menahu bagaimana proses pertukaran data dalam jaringan *cloud*, padahal *cloud* sangat rentan terhadap berbagai ancaman keamanan dan pihak *provider* seringkali memilih menyembunyikan kelemahan itu, fenomena itu yang menimbulkan keraguan dan ketidak percayaan pengguna atas teknologi *cloud*, kekhawatiran berdampak pada keamanan data yang disimpan pada data center jika terintegrasi dengan *cloud* computing, khususnya *cloud computing* yang bersifat public, oleh karena itu keamanan pada *cloud computing* menjadi salah satu masalah yang perlu diperhatikan sebagai salah satu strategi "pencegahan lebih penting dari pengobatan", dengan cara mendeteksi potensi sebuah celah keamanan dan memberikan peringatan pada *console* sedini mungkin lebih baik daripada merespon sebuah serangan setelah sistem sudah dalam keadaan kritis.

Cloud computing

Cloud computing memiki 6 komponen arsitektur, yaitu [3]:

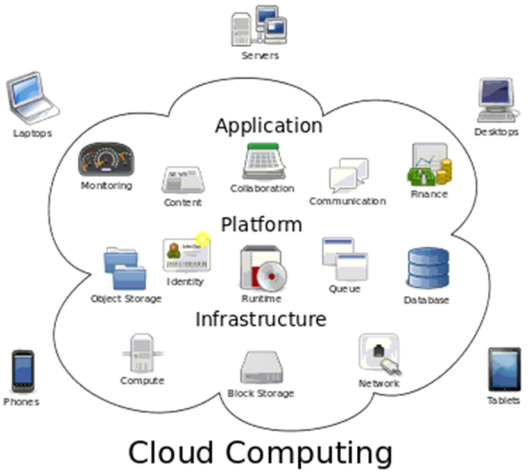
1. Client / End-User, yaitu pengguna yang memanfaatkan *web, application atau device front-end*.
2. Service, atau fungsi dari layanan cloud computing
3. *Application*, atau disebut *backbone* dari layanan *cloud computing.*
4. *Plaform atau infrastruktur software* untuk aplikasi *cloud computing.*
5. Storage, data center atau basis data penyimpanan dari cloud computing
6. Infrastruktur, yaitu komponen backbone dari cloud computing.



Gambar 2. Komponen dan Arsitektur *Cloud Computing*

Cloud computing juga memiliki 3 model kontrak layanan pada pengguna yaitu : [4]

1. *Software as a Service* (SaaS)
   Software as a Service merupakan evolusi lebih lanjut dari konsep ASP (*Application Service Provider*), sesuai namanya, SaaS memberikan kemudahan bagi pengguna untuk bisa memanfaatkan sumber daya perangkat lunak dengan cara berlangganan sehingga tidak mengeluarkan investasi, baik untuk *in house development* atau pembelian lisensi, dengan cara berlangganan *via web*, pengguna dapat langsung mengunakan berbagai fitur yang disediakan oleh penyedia layanan, dengan konsep SaaS ini pelanggan tidak memiliki kendali penuh atas aplikasi yang mereka sewa, tapi hanya mengendalikan fitur-fitur aplikasi yang telah disediakan oleh provider.
2. *Platform as a Service (PaaS)*
   Platform as a Service adalah layanan *cloud computing* yang menyediakan modul-modul siap pakai dan dapat digunakan untuk mengembangkan sebuah aplikasi, seperti halnya layanan SaaS, Pengguna PaaS tidak memiliki kendali terhadap sumber daya komputasi dasar seperti memori, media penyimpanan, *processing power* dan lain-lain, semuanya di atur oleh provider layanan PaaS, Pionir di area ini adalah *Google Apps Engine* yang menyediakan berbagai tools untuk mengembangkan aplikasi di atas platform *google* dengan mengunakan pemrograman *Phyton* dan *Django*.
3. *Infrastructure as a Service (IaaS)*
   Infrastructure as a Service adalah sebuah layanan yang "menyewakan" sumber daya teknologi informasi dasar yang meliputi media penyimpanan, *processing power, memory*, sistem operasi, kapasitas jaringan dan lain-lain. Dapat digunakan oleh penyewa untuk menjalankan aplikasi yang dimilikinya, Model bisnisnya mirip dengan penyedia *data center* yang menyewakan ruang untuk *co-location*, tapi ini lebih ke level mikronya. Penyewa tidak perlu tahu dengan mesin apa dan bagaimana cara penyedia layanan menyediakan layanan IaaS, yang penting permintaan mereka atas sumber daya dasar teknologi informasi itu dapat terpenuhi.



Gambar 1. Model Layanan *Cloud Computing*

Cloud Computing Security Issue

Cloud computing sendiri memiliki keuntungan dan kekurangan, keuntunganya adalah tak perlu lagi menyediakan infrastruktur atau perangkat penyimpanan data sendiri, kekuranganya adalah data-data disimpan di data center yang digunakan bersama-sama, hal ini menimbulkan potensi keamanan, diantaranya adalah resiko serangan terdistribusi seperti *Denial of Service* (DOS) atau *Distributed Denial of Service* (DDOS) yang menyerang penyedia layanan dan pengguna *cloud computing* sehingga melumpuhkan ketersediaan layanan *cloud* [5] dan mempengaruhi *Service Level Agreement* (SLA) antara pengguna dan penyedia layanan cloud. Untuk jenis serangan di atas di dapat diminimalisir dengan *passive system* atau *Intrusion Detection System* (IDS) sebagai penguatan mekanisme pertahanan dengan memberikan peringatan dini pada potensi serangan pada *cloud computing*.

Pada paper ini akan membahas sistem pendeteksi penyudup atau intrusion detection system barbasis mobile agent yang bisa di aplikasikan pada sisi *cloud client*, dimana *mobile agent* memiliki karakteristik yang bisa membantu memperbaiki efisiensi performa sistem pendeteksi penyusup atau *intrusion detection system* (IDS) di lingkungan *cloud computing*, penggunaan mobile agent pada intrusion detection system telah di ajukan pada beberapa penelitian [6][7][8]. Adapun dari penelitian tersebut tantangan IDS pada *cloud* network adalah skalabilitas, *latency*, *network load*, *mobility, false positive rate dan single point of failure*.

2. Pembahasan

2.1 Sistem Pendeteksi Intrusi (*Intrusion Detection System*)

Intrusion Detection System (IDS) merupakan *software* atau *hardware* yang melakukan otomatisasi proses monitoring kejadian yang muncul di sistem komputer atau jaringan dan menganalisanya untuk menemukan permasalahan keamanan. IDS adalah pemberi peringatan

dini pada administrator jika penyusup mencoba membobol sistem keamanan, kemudian administrator yang akan mengambil keputusan. Secara umum penyusupan bisa berarti serangan atau ancaman terhadap keamanan dan integritas data, serta tindakan atau percobaan untuk melewati sebuah sistem keamanan yang dilakukan oleh seseorang dari internet maupun dari dalam sistem. IDS tidak dibuat untuk menggantikan fungsi *firewall* karena kegunaanya sama sekali berbeda. Sebuah sistem firewall akan membatasi akses network dengan tujuan untuk mencegah penyusupan akan tetapi tidak bisa mengetahui apakah sebuah serangan sedang terjadi atau tidak. Dan IDS mengetahui dengan mengevaluasi sebuah potensi serangan dan memberi peringatan awal [9]. Dengan meningkatnya jumlah serangan pada sebuah *network*, IDS merupakan sesuatu yang diperlukan pada infrastruktur keamanan. Secara singkat, fungsi IDS adalah pendeteksi dan pemberi peringatan atas serangan yang terjadi pada sistem.

Akan tetapi dikarenakan infrastrukur *cloud* memiliki *network traffic* yang tinggi, IDS tradisional tidak cukup efisien untuk mengatasi *flow data* yang begitu besar, dibutuhkan *load balancing* IDS di environment *cloud computing* untuk mengatasi *data flow* yang besar, pada IDS tradisional memonitor, mendeteksi dan memberi peringatan pada administrator dengan mendeploy pada *key point machine* yang mengakibatkan *bottleneck pada network,,* akan tetapi dalam *cloud computing* harus di deploy di *cloud* server atau *virtual machine*, dalam kasus ini jika ada penyusupan, serangan atau pencurian data client, *cloud* user tidak akan diberitahukan secara langsung, dikarenakan provider layanan *cloud* akan khawatir akan reputasi.[10]

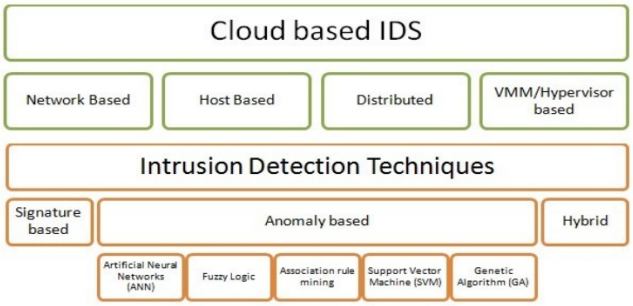
Tipe Intrusion Detection System

Ada 4 type *Intrusion Detection System* diataranya adalah adalah:

1. *Network-based* IDS (NIDS) mengidentifikasi penyusupan pada *key network point* untuk memonitor traffict yang masuk dan keluar dari semua device network, dan mencocokan traffict yang masuk dengan serangan yang sudah diketahui, jika dektahui aktifitas abnormal, selanjutnya akan memberikan peringatan pada administrator
2. *Host based IDS* (HIDS) memonitor secara spesifik pada individual *host-machine* atau *device network,* memonitor *packet* yang masuk dan keluar dari *host-machine* dan memberi peringatan administrator jika terdeteksi aktivitas network yang mencurigakan.
3. *distributed IDS (dIDS)* terdiri dari beberapa IDS di skala network yang besar, dIDS saling berkomunikasi dengan central server dan co-operative agent yang di distribusikan di seluruh network. ,
4. *IDS Virtual Machine Manager (VMM)* terdiri dari *aplikasi yanag dapat menjalankan beberapa operating system yang di pasang* IDS untuk memonitor dan mengaudit permintaan akses pada *hardware dan software pada cloud computing, IDS VMM memiliki mekanisme yang mirip seperti* HIDS.

Intrusion Detection System juga mengunakan 3 metode pendeteksian yaitu:

1. *Anomaly detection,* mendeteksi penyusup melalui pengenalan pola pada perilaku aatau *behavior* yang mencurigakan atau tindakan yang tidak normal. Deteksi anomali ini biasanya mengunakan metode *fuzzy logic, association rule mining, support vector machine, generic algorithm.*
2. *Signature detection*, mendeteksi penyusup berdasarkan *knowledge* karakteristik serangan pada rule yang sudah didefinisikan sebelumnya.
3. *Hybrid detection*, adalah kombinasi pendeteksian dari deteksi *signature* dan *anomal*



Gambar 2. Type IDS (Hijau), Metode Deteksi IDS (Orange)

IDS memberikan peringatan untuk administrator yang didasarkan pada *true positve* atau *true alarm* ketika benar-benar terjadi penyusupan atau serangan, dan *false positif* atau *false alarm* jika terjadi kesalahan deteksi pada sistem, IDS bisa mendeteksi pola serangan dengan cara mengawasi paket network, menerapkan *signature* dan memberikan peringatan pada *administrator. [11]*

2.2 Penelitian Terkait

Intrusion Detection System* pada *Cloud computing

Disamping problem kelemahan IDS yang disebutkan di atas, tantangan IDS pada *cloud* ke dapan yang perlu diperhatikan adalah skalibilitas dan beban network ketika layanan terus bertambah, mobilitas, intrusi / seranganskala besar/terdistribusi dan single point of falure yakni kegagalan sistem karena tidak adanya *fail over* / pengambil alihan service layanan jika salah satu node mati

Salah satu strategi IDS yang cukup *reliable* pada *cloud computing* environment adalah penerapan IDS pada tiap *virtual machine*, pada penelitian A. Bakshi, Yogesh B, [5] menyajikan dan mengevaluasi metode ini dengan perbandingan dengan pengembagan strategi IDS yang lainnya yang mengunakan single IDS pada cluster controller. IDS diterapkan pada tiap *virtual machine* pada *cloud computing platform* untuk menghilangkan permasalahan *overload data* dikarenakan *traffic network* pada semua engine IDS, penerapan IDS pada setiap *virtual machine* dapat menghilangkan masalah *overloaded* dikarenakan kebutuhan monitoring pada semua *traffic network* dari infrastruktur *cloud computing*. Akan tetapi, keterbatasan dari strategi IDS yang diterapkan pada tiap *virtual machine* masih missing link

pada fase korelasi yang mana disarankan untuk penelitian yang akan datang. Saran berikutnya disamping IDS untuk tiap virtual machine adalah menyertaan *Cloud Fusion Unit* (CFU) pada *front-end*, dengan tujuan mendapatkan dan mengendalikan alert yang diterima dari IDS sensor VM.

Pada penelitian A. Patel [12] mengusulkan sistem *cooperative* IDS (CIDS) untuk mendeteksi serangan DoS dalam jaringan *Cloud computing*, yang mana memiliki keuntungan mencegah sistem dari *single point of failure attack*, walaupun metode ini adalah solusi IDS yang memiki performa yang lebih lambat dari IDS berbasis *Snort*. Dengan demikian, *framework* yang sama diusulkan pada [13] adalah sistem IDS terdistribusi, dimana setiap IDS terdiri dari tiga modul tambahan: blok, komunikasi dan kerjasama, yang ditambahkan ke dalam sistem Snort pada bagaimana untuk mewujudkan sinkronisasi dan integrasi IDS pada Sensor VM.

Menerapkan IDS untuk setiap mesin virtual adalah ide yang juga disarankan pada penelitian J.H Lee [14], yang meningkatkan efektivitas IDS dengan menetapkan multi intrusion detection system dan analisis manajemen log dalam *cloud computing*. Dalam hal ini pengguna akan menerima tingkat keamanan yang tepat, yang akan ditekankan pada tingkat IDS diterapkan pada mesin virtual dan juga di tahap prioritas analisis log dokumen. Model keamanan multi-level ini kurang memecahkan masalah penggunaan sumber daya storage yang efektif.

Kemudian dilanjutkan pada penelitian V. Dastjerdi dan M. Uddin [6] yang mempunyai korelasi dengan mengimplementasi *Mobile Agent* pada IDS di tiap node, IDS menunakan metode *Bayession Classifier* untuk Analisis klasifikasi Fault-Tree pada IDS di setiap mesin virtual dan pada akhirnya hasil dari sensor akan menyatu dengan menggunakan kombinasi rule.

Metode lain yang mirip mengunakan multi IDS management log di ajukan dalam penelitian [14] yang mempresentasikan model teori IDS dalam *cloud computing*, dengan menggunakan IDS multi controller, yang menciptakan multi IDS untuk setiap pengguna. IDS ini dapat digunakan dalam beberapa kontroler Node dan Node kontroler dapat berisi IDS untuk setiap pengguna yang diterapkan pada *cloud* OS. Tahap analisis dari multi IDS untuk setiap pengguna berlangsung di controller IDS. Dibandingkan dengan penelitian S. Bharadwaja [15] di mana penekanannya adalah

yang antara lain berfungsi sebagai alokasi sumber komputasi IDS, kualitas layanan dan penanganan kegagalan sistem. Akan tetapi di koreksi oleh A. V. Dastjerdi, sendiri pada penelitiannya menyingung masalah pengendalian agent terdistribusi. Disharmoni antar agent berkaitan dengan pengendalian otonom agent yang mengakibatkan kegagalan sistem atau *single point of failure*.

Pada Tabel 1 akan memberi gambaran perbandingan penelitian terakait sebelumnya, dengan matrix perbandingan mobile agent dan mobile agent, teknik deteksi, tipe IDS, penempatan IDS, Tipe deteksi, data souce, attack coverage dan limitasi pada penelitian terkait.

Tabel 1: Matrix Perbandingan Penelitian Terkait

	Refrensi	Teknik Deteksi	Tipe IDS	Penempatan IDS	Tipe Deteksi	Data Source	Attack Coverage	Limitasi
Mobile Agent	*Distributed Intrusion Detection in Clouds using Mobile Agents[6]*	Signature Based	Distributed	Tiap Node	Real-Time	Network Traffict, Signature Serangan yang diketahui	Mendeteksi serangan yang diketahui	Mobile agent yang terbatas pada virtual machine
	Signature-based Multi-Layer Distributed Intrusion Detection System using Mobile Agents[7]		Distributed	Tiap Node	Real-Time	Network Packet, signature intrusi yang diketahui	Mendeteksi serangan yang diketahui	signature database yang terbatas
	A Distributed Intrusion Detection System Based on Mobile Agents[8]		Distributed	Tiap VM	Real-Time	Network Packet, signature intrusi yang diketahui	Mendeteksi serangan yang diketahui	Single Point of Failure
Non Mobile Agent	*Securing Cloud from DDOS using IDS in VM, 2010[5]*		Network Based	Virtual Switch	Real-Time	Network Packet, signature intrusi yang diketahui	Mengamankan VM dari DDOS attack	Mendeteksi serangan yg diketahui, Overload resource
	Multi IDS & log management, 2009[14]	Anomaly Based	Host Based	*cloud* OS	Real-Time	User behavior, pola serangan	Bisa mendeteksi serangan yg diketahui dan tidak diketahui	Overload resouce

	IDS cloud computing environment, 2011[10]	Hybrid Based	Host Based & Network Based	Tiap Node	Real-Time	Log user activities, signature intrusi yang diketahui	Bisa mendeteksi semua type serangan	Perlu pengujian yang lebih kompleks
	Bayession Classifier & Snort Based in NIDS in Cloud computing, 2012[13]		Network Based	Tiap Processing Server	Real-Time	Network Packet, signature intrusi yang diketahui	Bisa mendeteksi semua type serangan	Compleksitas akan bertambah dengan kombinasi teknik deteksi

Tantangan *IDS pada cloud computing*

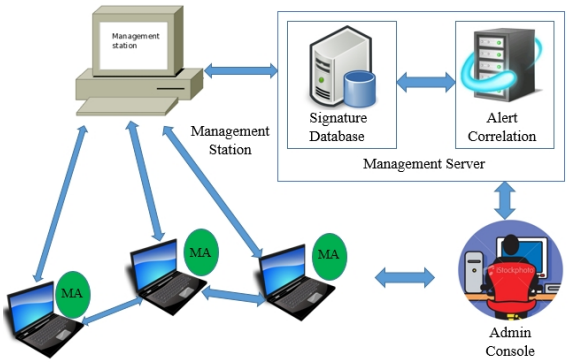
Pengembangan IDS pada *cloud computing* baik dengan pendekatan metode *anomaly* dan *signature detection* memiliki beberapa tantangan, antara lain, yaitu [11] :

1. *Variants,* Signature dibangun terus untuk memodelkan serangan serangan terbaru. Mengetahui sukses atau tidaknya sebuah set signature harus cukup unik untuk memberikan peringatan atau alert pada saat yang memang berbahaya. Kesulitannya adalah kode-kode untuk *exploit* itu dengan mudahnya dimodifikasi oleh penyerangnya, jadi sangat memungkinkan sekali banyak terjadi variasi pada kodenya.
2. *False Positives Rate,* yaitu merupakan alert yang memberitahu adanya aktifitas yang berpotensi berupa serangan, tetapi masih ada kemungkinan bahwa ternyata aktifitas tersebut bukan sebuah serangan. Kesulitannya adalah apabila jumlah alert atau peringatan begitu banyak dan sulit untuk menyaring mana yang memang benar-benar serangan atau bukan.
3. *False Negatives, yaitu* terjadi pada kondisi dimana IDS tidak dapat mendeteksi adanya serangan, karena tidak mengenal signature-nya. Jadi IDS tidak memberikan alert, walaupun sebenarnya serangan terhadap sistem tersebut sedang berlangsung.
4. *Scalability,* Skalabilitas network yang besar juga mendatangkan potensi ancaman keamanan baik intrusi atau serangan skala besar atau terdistribusi.
5. *Network Load / Traffic,* infrastrukur *cloud* memiliki *network traffic* yang tinggi, IDS tradisional tidak cukup efisien untuk mengatasi *flow data* yang begitu besar pda cloud computing
6. *Single Point of Failure,* penerapan IDS yang di tempatkan pada *virtual machine* atau *node core* akan memakan *resource* yang besar *pada* sistem, hal ini akan menyebabkan *bottleneck system* dan kegagalan sistem atau *single point of failure* IDS

2.3 Model yang di Usulkan

Oleh karena permasalahan di atas diajukan solusi metode *Real-time Cloud Intrusion Detection System* terdistribusi mengunakan *Mobile Agent* yang di distribusikan pada layer *environment Software as a Service* (SaaS) untuk membagi beban (*load balancing*). Mobile agent IDS ini diharapkan dapat mengidentifikasi serangan terdistribusi secara efektif, ada 5 komponen pada IDS berbasis mobile agent :

1. *Mobile Agent,* yaitu perangkat lunak berbasis agent yang di distribusikan pada sisi *cloud client* sebagai sensor untuk memonitor network dan aktivitas sistem dan diteruskan pada management station
2. *Management station*, mengintergrasikan informasi dari *mobile agent* sebagai IDS *distributed sensor pada cloud computing* , memberikan *view* yang *komprehensif* dan mudah di pahami, bisa berupa *web front-end* atau *dedicated graphical user interface.*
3. *Management Server, Signature Database,* yaitu basis data pengenalan pola serangan, IDS berbasis siganture mempunyai keuntungan yaitu tingkat *false alarm* dan pengunaan resouce sistem yang lebih kecil di bandingkan IDS berbasis *anomaly.*
4. *Management Server, Alert Correlation,* yaitu suatu proses dalam *management server* yang menganalisa *alert* / peringatan yang di hasilkan sistem deteksi intrusi.
5. *Admin Console,* adalah point central management dari IDS, dengan console ini administrator dapat memonitor setiap peringatan serangan. Sebelum di sajikan pada user client dengan *graphical user interface alert.*



Gambar 3: Alur kerja dan arsitektur solusi yang diusulkan

Adapun standar teknologi yang digunakan dalam usulan penelitian adalah seagai berikut :

1. *Apache cloud stack*, sebagai platform mendeploy dan memanage virtual machine cloud berbasis open source [16]
2. *Java*, sebagai plaform software application dalam pengembangan IDS berbasis mobile agent
3. *Jade*, *java agent development framework* untuk pengembanagan *mobile agent*
4. *Snort & Suricata*, sebagai IDS *engine dan network monitoring* berbasis *opensource* [17]

3. Kesimpulan

Berdasarkan pembahasan diatas, dapat ditarik kesimpulan beberapa metode *intrusion detection* memang lebih aman dan lebih efisien jika dibandikan dengan metode *intrusion detection system* traditional, akan tetapi untuk penerapan pada *cloud computing* perlu dikaji lebih dalam lagi, dikarenakan infatrukur *cloud* memiliki *network traffic* yang sangat tinggi, begitu juga penerapan IDS yang di tempatkan pada *virtual machine* atau *node core* akan memakan *traffic network rate* dan *resource* yang besar *pada* sistem, hal ini akan menyebabkan *bottleneck system* dan kegagalan sistem atau *single point of failure* IDS. meskipun cukup dapat mengatasi kelamahan IDS dengan metode tersebut tidak cukup efisien untuk mengatasi *flow data* yang begitu besar.

IDS tidak dibuat untuk menggantikan fungsi firewall. firewall tidak bisa mengetahui apakah sebuah serangan terjadi atau tidak sedangkan IDS mengetahuinya. singkatnya, fungsi IDS adalah pemberi peringatan atas serangan yang terjadi pada sistem. Akan tetapi IDS tidak bisa melakukan investigasi serangan tanpa campur tangan manusia, oleh karena itu IDS perlu pengembangan lebih lanjut dan adaptasi pada teknologi yang sedang berkembang.

Cloud computing mempunyai potensi ancaman keamanan yg lebih besar dari pada network tradisional karena serangan terdistribusi pada *cloud* akan menimbulkan masalah *network access rate* dan *control data/privacy* dan ketersediaan penyedia layanan *cloud*, sehingga pada usulan model solusi IDS *berbasis mobile agent* diharapkan akan menangani masalah *load balancing* dan *high avaibility intrusion detection system, sehingga IDS bisa dijadikan software as service (SaaS) pada pengguna cloud computing*.

Daftar Pustaka

[1] Purbo, Onno W. 2012. *Membuat Sendiri Cloud Computing Server Mengunakan Open Source". Yogyakarta: CV. Andi Offset*
[2] Pangsa Pasar Pengguna Tumbuh 66,67% « *Cloud* for SME and Enterprise Telkom*Cloud*." [Online]. Available: http://www.telkomcloud.com/sme/pangsa-pasar-pengguna-tumbuh-6667/. [Accessed: 25-Nov-2014]
[3] Wahana, Komputer, 2011. Kupas Tuntas Bermacam Aplikasi Generasi Cloud Computing, *Yogyakarta: CV. Andi Offset*
[4] Tim Elcom, 2012, Cloud Computing – Aplikasi berbasis web yang menguvah cara kerja dan kolaborasi anda secara online, *Yogyakarta: CV. Andi Offset.*
[5] A. Bakshi, Yogesh B, "Securing *cloud* from DDOS Attacks using Intrusion Detection System in Virtual Machine", 2010 Second International Conference on Communication Software and Networks, pp. 260-264.
[6] A. V. Dastjerdi, K. A. Bakar, S. G. H. Tabatabaei, "Distributed Intrusion Detection in *Cloud*s using Mobile Agents", Third International Conference on Advanced Engineering Computing and Applications in Sciences, 2009, pp. 175-180.
[7] M. Uddin, A. A. Rehman, N. Uddin, et al., "Signature-based Multi-Layer Distributed Intrusion Detection System using Mobile Agents", International Journal of Network Security, Vol. 15, No. 1, Jan. 2013, pp. 79-87.
[8] M. Xiu-liang, W. Chun-dong, W. Huai-bin, "A Distributed Intrusion Detection System Based on Mobile Agents", IEEE 2009.
[9] A. Razzaq, A. Hur, S. Shahbaz, M. Masood, and H. F. Ahmad, "Critical analysis on web application firewall solutions," in 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 2013, pp. 1–6.
[10] S. N. Dhage, B. B. Meshram, R. Rawat, S. Padawe, M. Paingaokar, A. Misra , "Intrusion Detection System in *Cloud computing* Environment", International Conference and Workshop on Emerging Trends in Technology (ICWET 2011), pp. 235-239.
[11] Yasir Mahmood, Umme Habiba "Intrusion Detection System in Cloud Computing: Challange and Opportunity" in 2013 2nd National Converenceon Information Assurance, 1-8
[12] A. Patel, Q. Qassim, Z. Shukor, J. Nogueira, J. Júnior and C. Wills, "Autonomic Agent-Based Self-Managed Intrusion Detection and Prevention System", Proceedings of the South African Information Security Multi-Conference (SAISMC 2010), pp. 223-234.
[13] C. N. Modi, D. R. Patel, A. Patel, R. Muttukrishnan, "Bayesian Classifier and Snort based Network Intrusion Detection System in *Cloud computing*", Third International Conference on Computing, Communication and Networking Technologies, 26th-28th July 2012.
[14] J. H. Lee, M. W. Park, J. H. Eom, T. M. Chung, "Multi-level Intrusion Detection System and Log Management in *Cloud computing*", ICACT, 2011, pp. 552-555.
[15] S. Bharadwaja, W. Sun, M. Niamat, F. Shen, "Collabra: A Xen Hypervisor based Collaborative Intrusion Detection System", Eighth International Conference on Information Technology: New Generations, 2011, pp. 695-700.
[16] Suricata: The Snort Replacer (Part 1: Intro & Install), Jul 24, 2013, http://www.linux.org/threads/suricata-the-snort-replacer-part-1-intro-install.4346/
[17] http://mail-archives.apache.org/mod_mbox/cloudstack-users/201311.mbox/browser *cloud*stack-users mailing list archives: November 2014,

Biodata Penulis

Fahmi Dzikrullah, memperoleh gelar Sarjana Komputer (S.Kom.), Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang, lulus tahun 2013. Saat ini menjadi peneliti di *e-Systems* Lab dan mahasiswa Program Pascasarjana di Jurusan Teknik Elektro dan Teknologi Informasi UGM. Bidang-bidang penelitian yang diminati meliputi *Web Development, Network & Security.*

Selo Sulistyo, memperoleh gelar Ph.D. di bidang Information and Communication Technology dari University of Agder, Norway, lulus tahun 2012. Saat ini menjadi kepala e-Systems Lab dan dosen di Jurusan Teknik Elektro dan Teknologi Informasi UGM. Bidang-bidang penelitian yang diminati meliputi Model-driven Software Engineering, Software Development, serta Mobile and Embedded Programming in the Internet of Services.

Noor Akhmad Setiawan, memperoleh gelar Ph.D. dari Universiti Teknologi Petronas Malaysia, lulus tahun 2009. Saat ini menjadi kepala grup riset intelligent-Systems (*i-sys*) Lab dan dosen di Jurusan Teknik Elektro dan Teknologi Informasi UGM. Bidang-bidang penelitian yang diminati meliputi Artifical Intelligent dan Datamining