

RANCANG BANGUN SISTEM INFORMASI PERHOTELAN BERBASIS KOMPUTASI AWAN

Ence Achmad A.A.P.¹, Banni Satria A.², Yan Watequlis S.³

Program Studi Teknik Informatika, Teknik Elektro, Politeknik Negeri Malang
¹encekertosastro@gmail.com, ²banniandoko@gmail.com, ³yan_ws@yahoo.com

Abstrak

Sistem Informasi Perhotelan Konvensional seringkali diimplementasikan secara *on-premise* dimana pengguna dihadapkan permasalahan seperti tingginya biaya pengadaan infrastruktur hingga pemborosan sumber daya yang diperlukan untuk melakukan instalasi, *maintenance* dan *versioning* sistem. Kehadiran dari *Cloud Computing* telah menjadi segmen yang bertumbuh sangat cepat dalam industri informasi dan teknologi. *Cloud Computing* menawarkan karakteristik-karakteristik yang unik untuk bisa mengatasi masalah-masalah yang biasa dihadapi pada penerapan sistem informasi konvensional. Pada penelitian ini, penulis menerapkan *Cloud Computing* dengan model *Software as a Service* pada Sistem Informasi Perhotelan untuk mengetahui apakah *Cloud Computing* dengan model *Software as a Service* dapat mengatasi permasalahan yang muncul pada penerapan Sistem Informasi Perhotelan Konvensional.

Kata kunci : Cloud Computing, Hotel Management System

1. Pendahuluan

Sistem informasi perhotelan adalah sebuah program komputer (*hotel management software*) yang bertujuan untuk membantu manajemen hotel, baik dalam kegiatan manajemen sehari-hari maupun dalam menghasilkan laporan-laporan yang diperlukan hotel. Dari sisi manajemen, kegiatan itu meliputi pengelolaan kamar, harga, fasilitas hingga staff. Dari sisi *front-office*, kegiatan meliputi pengelolaan tamu, *housekeeping*, reservasi hingga *check in/out*. Dan dari sisi customer seperti reservasi online.

Komputasi Awan (*Cloud Computing*), sebuah visi yang terpendam lama atas komputasi sebagai alat, memiliki potensi untuk mengubah bagian besar dari industri Informasi dan Teknologi. Merujuk pada McKinsey & Company yang mendasarkan 22 artikel penelitian untuk mendefinisikan *Cloud Computing* (Forrest, 2009), *Clouds* adalah layanan yang berbasis *hardware* yang memberikan komputasi, jaringan dan *storage* dimana: arsitektur dan manajemen *hardware* diabstraksikan dan dipisahkan dari pembeli, pembeli dikenakan biaya infrastruktur yang berbasis *pay-per-use* dan skala kapasitas infrastruktur dapat diatur secara elastis.

Sedangkan, Layanan *Cloud (Cloud Services)* hanya perlu memenuhi poin pertama dan ketiga (Arsitektur dan manajemen *Hardware* diabstraksikan, dipisahkan dari pembeli dan skala kapasitas infrastruktur dapat diatur secara elastis). Terdapat tiga model penyampaian *Cloud Computing*, yaitu: *Software as a Service (SaaS)*, *Platform as a Service (PaaS)* dan *Infrastructure as a Service (IaaS)*.

Dengan Komputasi Awan pengembang tidak lagi harus khawatir mengenai pengadaan infrastruktur yang berlebihan ketika layanan mereka ternyata tidak sesuai dengan prediksi, sehingga membuang-buang sumberdaya, atau mungkin sebaliknya. Selain itu, kelebihan dari *Software as a Service (SaaS)* untuk pengguna maupun penyedia layanan diantaranya: Penyedia layanan mendapatkan kesederhanaan instalasi, *maintenance* dan *versioning* dengan tersentralisasi. Pengguna dapat mengakses layanan ini “dimanapun dan kapanpun”, berbagi data, kolaborasi dengan mudah dan menyimpan data mereka dengan aman pada infrastruktur.

1.1 Clouds

Clouds adalah layanan yang berbasis hardware yang memberikan komputasi, jaringan dan *storage* dimana: arsitektur dan manajemen *hardware* diabstraksikan dan dipisahkan dari pembeli, pembeli dikenakan biaya infrastruktur yang berbasis *pay-per-use* dan skala kapasitas infrastruktur dapat diatur secara elastis (Forrest 2009).

Berdasarkan National Institute of Standard and Technology (NIST) sebuah Sistem dapat dikatakan *Cloud Computing System* harus memenuhi lima esensi sebagai berikut:

a. *On-demand self-service*

Layanan cloud yang berupa aplikasi web dapat diatur secara otomatis sesuai kebutuhan oleh pengguna tanpa membutuhkan interaksi dari pihak penyedia.

b. *Broad Network Access*

Pengguna dapat mengakses layanan *cloud* melalui internet kapanpun dan dimanapun.

c. *Resource Pooling*

Sumberdaya fisik maupun virtual terhimpun di dalam *cloud* dimana pengguna aplikasi tidak memiliki kontrol maupun pengetahuan dasar atas sumberdaya.

d. *Rapid Elasticity*

Pengadaan skala kapasitas atas sumberdaya dan infrastruktur dapat diatur secara elastis dan cepat.

e. *Measured Service*

Penggunaan sumberdaya *cloud* dan layanan dimonitor, dikontrol dan dioptimalkan oleh penyedia layanan melalui model bisnis *pay-per-use*.

1.2 Software as a Service

Software as a Service adalah salah satu tipe dari *Cloud Services* dimana memberikan layanan *software* via internet (Sarbojit et al. 2014). *Software as a Service* dapat dideskripsikan sebagai sebuah proses dimana *Application Service Provider (ASP)* menyediakan aplikasi melalui internet dengan pendayagunaan infrastruktur *Cloud* dengan model pematokan harga *pay-as-you-go*. Hal ini membuat pelanggan tidak perlu lagi mengkhawatirkan mengenai instalasi dan pengoperasian aplikasi pada komputer mereka. Sebaliknya, penyedia *SaaS* memegang penuh semua kewajiban untuk memmanage infrastruktur (*server, operating system, database, data center, network* dan sebagainya). Selain itu penyedia *SaaS* juga bertanggung jawab atas *patch* atau *upgrade* aplikasi (Leena et al. 2010).

Software as a Service dibangun diatas tiga *layer* utama, yaitu: *infrastructure layer, application platform layer* dan *application layer*. *Infrastructure Layer* terdiri atas sumberdaya komputasi dan jaringan (*data center, hardware, servers, application programming interface (API)*). Integrasi dan interaksi antara infrastruktur dan aplikasi ditangani *layer* kedua, yaitu *Application Platform Layer* yang berupa sistem operasi, server aplikasi, bahasa *programming* dan sebagainya. Terakhir adalah *application layer* yang mencakup pengembangan sebenarnya, komersialisasi dan penyampaian aplikasi (Churakova et al. 2010).

1.3 Multi-tenancy

Cloud service dibangun didalam lingkungan multi-tenant. Hal ini berarti para pengguna menggunakan aplikasi yang sama, menggunakan sistem operasi yang sama pada hardware yang sama dengan data storage sama. Pemisahan pengguna dicapai didalam desain aplikasi sehingga pengguna tidak berbagi atau melihat data satu sama lain (Multitenancy, 2010).

Perbedaan utama *Software as a Service* dengan on-premise software adalah fakta dimana semua data pengguna digunakan dan diproses melalui penggunaan aplikasi dan disimpan pada database

penyedia (Churakove et al. 2010). Terdapat tiga pilihan arsitektur data dalam konteks multi-tenant:

a. *Separate Database*

Penyedia layanan memberikan setiap pengguna *database* terisolasi dimana tidak dapat diakses oleh pengguna lain.

b. *Shared Database, Separate Schema*

Setiap pengguna memiliki atribut unik pada *schema* yang terisolasi namun berada pada database yang sama dengan pengguna lain.

c. *Shared Database, Shared Schema*

Setiap pengguna memiliki *tenant-id* yang digunakan untuk mengakses data pada *schema* dan *database* yang sama dengan pengguna lain.

2. Metodologi Penelitian

Pada penelitian ini, penulis menggunakan beberapa metode penelitian sebagai berikut:

2.1 Pengumpulan Data

Pengumpulan data yang diperlukan untuk penelitian ini menggunakan metode-metode sebagai berikut:

a. Observasi

Penulis melakukan pengamatan langsung Sistem Informasi Perhotelan yang digunakan pada Hotel Ibis Malioboro. Sistem Informasi perhotelan ini memiliki fitur-fitur yang mencakup proses-proses dari *front-office, housekeeping* hingga *management* hotel. Observasi ini dilakukan untuk mendapatkan fitur-fitur esensial yang ada pada Sistem Informasi Perhotelan sehingga dapat menentukan rancangan pengembangan sistem baru yang akan dibangun agar dapat memenuhi kebutuhan pengguna Sistem Informasi Perhotelan.

Selain pengumpulan data kebutuhan sistem, pada langkah ini penulis juga mengumpulkan data-data yang diperlukan untuk melakukan pembangunan aplikasi, seperti: *sample* data fasilitas, tamu hingga reservasi. Selain pengambilan *sample* data, penulis juga melakukan pengumpulan *sample* format laporan-laporan yang ada pada Sistem Informasi Perhotelan ini.

b. Studi Pustaka

Pada tahapan pengumpulan data dengan cara studi pustaka, penulis melakukan pencarian referensi-referensi yang relevan dengan objek yang akan diteliti. Pencarian referensi dilakukan pada perpustakaan hingga melalui internet.

Setelah mendapatkan referensi-referensi yang relevan tersebut, penulis melakukan pencarian informasi-informasi yang dibutuhkan dalam melakukan penelitian ini. Hasil dari pencarian informasi-informasi tersebut digunakan penulis untuk menyusun landasan teori, metodologi penelitian serta perancangan

aplikasi.Pustaka-pustaka yang dijadikan acuan dapat dilihat pada bagian Daftar Pustaka.

2.2 Pengembangan Sistem

Pada penelitian ini, penulis menggunakan metode pengembangan sistem SDLC model Waterfall. Pemilihan model Waterfall ini dilakukan dengan alasan proyek yang dibangun memiliki scope yang tidak berubah-ubah, system requirement tidak ambigu, serta proyek dilakukan dalam waktu yang singkat. Kelebihan-kelebihan model Waterfall diantaranya:

- Model sederhana dan mudah untuk dimengerti dan digunakan.
- Proses manajemen proyek mudah, pengaturan tugas lebih mudah.
- Fase diproses dan diselesaikan satu persatu.
- Bekerja baik dalam proyek dimana kebutuhan sistem jelas.
- Proses dan hasil didokumentasikan dengan baik.

Tahapan-tahapan yang dilakukan oleh penulis dalam melakukan pengembangan aplikasi pada penelitian ini adalah:

- Requirement Gathering and Analysis**
Pada tahap Requirement Gathering and Analysis penulis melakukan pengamatan pada Sistem Informasi Perhotelan pada Hotel Ibis Malioboro. Pengamatan sistem meliputi pengamatan fitur-fitur esensial yang ada, pengamatan data sample data serta sample format laporan yang ada. Selain itu, penulis juga menentukan alur bisnis dari aplikasi yang akan dibangun pada sistem yang dikembangkan.
- System Design**
Setelah melakukan pengumpulan dan analisa kebutuhan penulis melakukan desain sistem menggunakan metode konseptual dengan UML (Unified Modelling Language), pada konsep rancangan desain sistem menggunakan Use Case Diagram, Class Diagram dan Sequence Diagram. Untuk rancangan desain database penulis menggunakan Entity Relational Database dan Physical Data Model Diagram. Selain itu, perancangan yang dilakukan juga meliputi arsitektur layer infrastruktur cloud yang akan digunakan serta representasi antar-muka aplikasi.
- Implementation**
Pada tahap ini, rancangan yang telah dilakukan diimplementasikan secara langsung. Dalam penelitian ini rancangan database yang telah dibuat dibangun menggunakan MySQL. Untuk pengembangan aplikasi proses coding menggunakan bahasa PHP. Pengembangan dalam lingkungan local menggunakan Apache Server yang terdapat

dalam paket XAMPP 1.7.7 dan Amazon Web Service (AWS) OpsWorks dalam lingkungan cloud.

- Testing**
Tahapan setelah proses implementasi adalah proses testing dimana ditujukan untuk membuktikan validitas fungsional daripada sistem yang telah dibuat. Dalam tahap ini, penulis melakukan pengujian pada sistem menggunakan metode Black Box Testing. Black Box Testing dilakukan pada setiap fitur yang ada dengan menggunakan skenario-skenario yang dijelaskan pada bab pengujian. Selain itu, pengujian multi tenancy, maintenance dan versioning juga dilakukan dengan menjalankan skenario yang dijelaskan lebih lanjut pada bab pengujian.
- Deployment of System**
Tahapan ini merupakan peluncuran sistem dimana dilakukan setelah semua tahap-tahap sebelumnya diselesaikan dan telah ditentukan bahwa sistem yang dibangun telah laik untuk digunakan dalam lingkungan produksi. Adanya Dokumentasi versi release seperti instruksi penggunaan merupakan syarat yang harus dipenuhi dalam fase ini.
- Maintenance**
Setelah sistem diluncurkan kepada para pengguna, sistem harus selalu dalam kondisi terjaga. Jika ada pengguna menemukan masalah pada sistem, pengguna dapat melaporkan pada penyedia sehingga penyedia dapat melakukan maintenance atau update pada sistem. Fase ini adalah fase dimana dilakukannya perbaikan-perbaikan pada bug-bug yang ditemukan.

3. Hasil

Setelah setiap tahap pada metodologi penelitian dilakukan, didapatkan hasil seperti berikut.

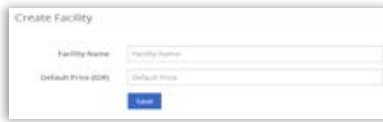
3.1 Fitur

- Manage Tenant

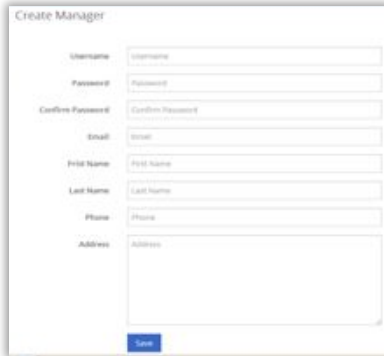


Gambar Detail Tenant

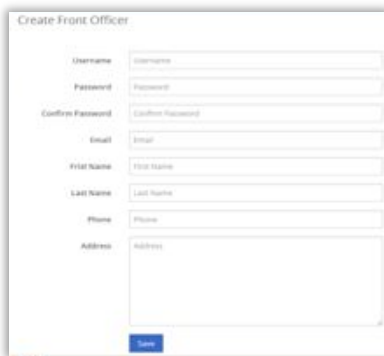
b. Manage Hotel Data



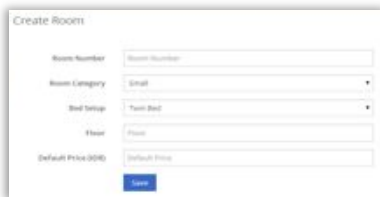
Gambar Create Facility



Gambar Create Manager



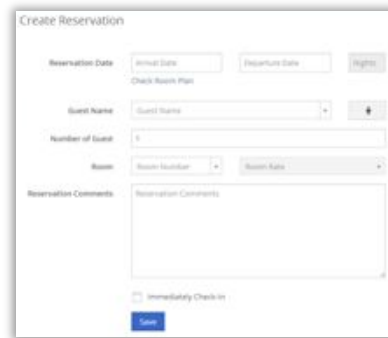
Gambar Create Manager



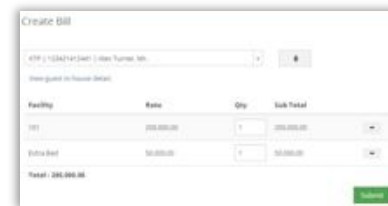
Gambar Create Room



Gambar Create Guest



Gambar Create Reservation



Gambar Create Bill

c. Online Reservation

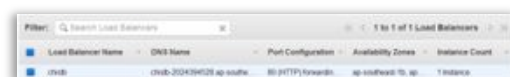


Gambar Online Reservation

3.2 Infrastruktur Cloud

a. Load Balancing Layer

Load Balancing Layer berfungsi untuk mendistribusikan lalu lintas masuk aplikasi pada sistem. Adapun implementasi instansi load balancer sistem sebagai berikut.



Gambar Load Balancer

Load Balancer sistem bernama “chislb” menjadi penghubung antara lalu lintas yang masuk pada sistem dan instansi server aplikasi PHP.

Instance ID	Name	Availability Zone	Status	Actions
i-41006809	ChisStack-php-app1	ap-southeast-1a	InService	Remove from Load Balancer

Gambar Load Balancer

b. PHP App Server Layer

PHP App Server Layer berfungsi sebagai server aplikasi PHP. Adapun implementasi instansi PHP App Server Layer sistem sebagai berikut.



Gambar PHP App Server Layer

Instansi *PHP App Server Layer* bernama “php-app1” yang memiliki spesifikasi sebagai berikut.

Instance php-app1	
Details	
Status	Running
Hostname	php-app1
Layers	PHP App Server
EC2 Instance ID	i-41006809
OpsWorks ID	a2b6c8df-f5a7-4834-b0ba-09c8ec6053
Instance type	t2.micro
Size	t2.micro
Subnet	subnet-a0d125c5-172.31.0.0/20 - ap-southeast-1a
Operating system	Amazon Linux
Architecture	64bit
Virtualization type	paravirtual
EBS Optimized	no
Root device type	EBS backed
Root device ID	vol-fba1195

Gambar PHP App Server Layer

c. Database Layer

Database Layer berfungsi sebagai pusat database yang digunakan pada seluruh sistem sebagai berikut.

Instansi database sistem menggunakan Relational Database Service (RDS) dengan spesifikasi sebagai berikut.

Settings	Configuration details
Instance Identifier: chisdb	DB Engine: mysql (5.6.17)
Address: chisdb-c7zgfakexv-ap-southeast-1.rds.amazonaws.com	License Model: general-public-license
Port: 3306	DB User: chisroot
Arn: arn:aws:rds:ap-southeast-1:409111930642:db:chisdb	DB Password: Protected value
Status: Available	Option Groups: default:mysql-5-6-encrypted
Apps: CHS	Parameter Groups: default:mysql-5-6-encrypted

Gambar Database Layer

4. Pembahasan

Berdasarkan analisis dan perancangan yang dilaksanakan pada sistem informasi perhotelan berbasis komputasi awan dapat menghasilkan sistem informasi hotel independen untuk setiap pengguna sistem informasi perhotelan. Berikut merupakan pembahasan penelitian.

4.1 Multi Tenant

Pengujian *multi tenant* pada sistem ditujukan untuk membuktikan validitas kemampuan dari sistem dan infrastruktur yang telah dibangun untuk menangani sejumlah pengguna secara simultan. Dengan adanya konsep *multi tenant* pengguna tidak perlu melakukan pengadaan infrastruktur. Selain itu, pengujian ini turut ditujukan untuk membuktikan validitas isolasi data antara pengguna satu dengan pengguna yang lain.

Pengujian ini dilakukan dengan cara membuat dua hotel melakukan penggunaan fitur-fitur sistem secara bersamaan. Untuk menirukan aktifitas-aktifitas manajemen hotel yang dilakukan pada sistem, penulis menggunakan *Selenium*.

Pengujian ini menguji fitur *create reservation* dimana berfungsi untuk membuat reservasi pada hotel pengguna sistem, *check-in* dimana berfungsi untuk mengubah status reservasi menjadi *resident*, *check-out* dimana berfungsi untuk mengubah status *resident* menjadi *check-out* dan *billing* yang berfungsi untuk membuat tagihan. Adapun spesifikasi data *reservasi* dari kedua hotel sebagai berikut.

Field	Hotel 1	Hotel 2
Arrival Date	2014-08-02	2014-08-02
Departure Date	2014-08-04	2014-08-04
Guest Name	KTP: 321654 guest one, Mr.	KTP: 321654 guest three, Mr.
Number of Guest	1	1
Room Number	101	103
Room Rate	IDR 200,000.00 [Default Price]	IDR 200,000.00 [Default Price]
Immediately Check-In	No	No
Arrival Date	2014-08-02	2014-08-02
Departure Date	2014-08-04	2014-08-04
Guest Name	KTP: 321654 guest two, Mrs.	KTP: 321654 guest four, Mrs.
Number of Guest	1	1
Room Number	102	104
Room Rate	IDR 250,000.00 [Default Price]	IDR 250,000.00 [Default Price]
Immediately Check-In	Yes	Yes

Gambar Tampilan Data Reservasi

Adapun hasil setelah kedua data hotel dimasukkan pada sistem secara bersamaan.

Room Number	Guest	Arrival	Departure	Category
101	guest one, Mr.	2014-08-02	2014-08-04	Medium
102	guest two, Mrs.	2014-08-02	2014-08-04	Medium

Gambar Laporan Reservation Hotel 1

Room Number	Guest	Arrival	Departure	Category
103	guest three, Mr.	2014-08-02	2014-08-04	Small
104	guest four, Mrs.	2014-08-02	2014-08-04	Small

Gambar Laporan Reservation Hotel 2

Adapun hasil setelah kedua data hotel dilakukan *check-out* dan pembayaran pada sistem secara bersamaan.

Date	* Guest Name	Total
2014-08-02	guest two, Mrs.	IDR 500,000.00

Gambar Daftar Billing Hotel 1

Date	* Guest Name	Total
2014-08-02	guest four, Mrs.	IDR 500,000.00

Gambar Daftar Billing Hotel 2

Berdasarkan gambar diatas, menunjukkan bahwa hotel1 berhasil membuat reservasi untuk *Mr. Guest One, Mrs. Guest Two* dan hotel2 berhasil membuat reservasi untuk *Mr. Guest Three, Mrs. Guest Four*. Setelah itu, *check-in Mrs. Guest Two* dan *Mrs. Guest Four* dilakukan *check-out* dan *create billing*. Hal ini membuktikan bahwa sistem dapat menjaga isolasi data *revesation* pada hotelpengguna sistem satu sama lain dengan baik

Daftar	Konvensional			Komputasi Awan
	Tahun 1	Tahun 2	Tahun 3	Tahun 1-3
Sistem Informasi Perhotelan	135,000,000.00	-	-	Termasuk
Sistem Operasi	11,000,000.00	-	-	Termasuk
Pengantar Tech Support	60,000,000.00	60,000,000.00	60,000,000.00	Termasuk
Hardware	31,400,000.00	-	-	Termasuk
Maintenance	11,000,000.00	13,800,000.00	13,800,000.00	Termasuk
Total/Tahun	248,400,000.00	73,800,000.00	73,800,000.00	13,200,000.00
Total	-	-	396,000,000.00	39,600,000.00
Biaya Rata-Rata Per Bulan (3 Tahun)	11,000,000.00	11,000,000.00	11,000,000.00	1,100,000.00

Pada tabel diatas, menjelaskan skenario penggunaan sistem informasi perhotelan selama tiga tahun dengan model konvensional dan model komputasi awan. Dengan sistem ini, pengguna dapat mengeliminasi biaya tinggi yang diperlukan untuk melakukan pengadaan infrastruktur dan pemborosan sumber daya yang dibutuhkan untuk melakukan instalasi, maintenance dan versioning sistem.

5. Kesimpulan dan Saran

Setelah melakukan analisa perancangan dan implementasi pembangunan Sistem Informasi berbasis Komputasi Awan maka dihasilkan kesimpulan dan *saran* sebagai berikut.

5.1 Kesimpulan

Berdasarkan perancangan, implementasi, pengujian dan pembahasan yang telah dijelaskan

pada bagian sebelumnya, maka diperoleh *kesimpulan* bahwa Sistem Informasi Perhotelan berbasis Komputasi Awan yang telah dibangun dapat mengeliminasi tingginya biaya pengadaan infrastruktur dan pemborosan sumber daya yang diperlukan untuk melakukan instalasi, *maintenance* dan *versioning* pada penerapan Sistem Informasi Perhotelan Konvensional.

5.2 Saran

Pengembangan Sistem Informasi Perhotelan berbasis Komputasi Awan selanjutnya, dapat dikembangkan sistem yang mengacu pada konsep arsitektur *RESTful Web Service* dan *Single Page Application* demi meningkatkan efektifitas, efisiensi, mobilitas sistem dan pengalaman pengguna.

6. Daftar Rujukan:

- Ahmed, E.Y. (2012): *Exploring Cloud Computing Services and Applications*, Journal of Emerging Trends In Computing and Information Sciences, Vol 3, No. 6.
- Banerjee, S. & Jain, S. (2014): *A Survey on Software as a Service (SaaS) Using Quality Model in Cloud Computing*, International Journal of Engineering and Computer Science, Vol 3 Issue 2014, pp 3598-3602.
- Bhardwaj, S., Jain, L. & Jain, S. (2010): *An Approach for Investigating Perspective of Cloud Software as a Service (Saas)*, International Journal of Computer Application, Vol 10, No 2.
- Churakova, I. & Mikhranova, R (2010): *Software as a Service: Study and Analysis of SaaS Business Model and Innovation Ecosystems*
- Duan, J., Faker, P., Fesak, A. & Stuart, T. (2012): *Benefits and Drawbacks of Cloud-Based Versus Traditional ERP Systems*.
- Forrest, W. (2009): *Clearing the Air on Cloud Computing*. Discussion Document from McKinsey and Company.