

Analisis dan Evaluasi Atribut Kualitas *Software Architecture* Sistem RFID Universitas Telkom dengan *Architecture Tradeoff Analysis Method* (ATAM)

Nabiel Shidqi¹, Dana S. Kusumo, S.T, M.T, PhD², Sri Widowati Ir., MT³

^{1,2,3}Teknik Informatika, Universitas Telkom; Jl. Telekomunikasi, Dayeuhkolot, Sukapura, Dayeuhkolot, Bandung, Jawa Barat 40257

Abstrak

Perkembangan teknologi identifikasi sudah sangat pesat. Saat ini muncul teknik identifikasi RFID yang berbasis gelombang radio untuk identifikasi barang, hewan, hingga manusia. Teknologi RFID digunakan diberbagai bidang kehidupan manusia, salah satunya pendidikan. Sama halnya dengan sistem lainnya, teknologi RFID juga memiliki arsitektur sistem. Universitas Telkom menerapkan teknologi ini untuk melakukan sistem absensi perkuliahan mahasiswa. Dalam penerapannya terdapat beberapa kendala pada atribut kualitas sistem. Sehingga dibutuhkan analisis dan evaluasi terhadap atribut kualitas arsitektur system absensi tersebut agar menjadi lebih baik kedepannya. Penelitian ini menggunakan metode *Architecture Tradeoff Analysis Method* (ATAM) untuk melakukan evaluasi atribut kualitas arsitektur system. Metode ini menggunakan scenario dari tiap atribut kualitas yang menjadi acuan untuk analisis pendekatan arsitektur system. Atribut kualitas yang dianalisis adalah *reliability*, *performance* dan *usability*. Hasil dari analisis ini berupa poin sensitivitas, tradeoff dan resiko arsitektural dari masing-masing scenario atribut kualitas yang digunakan sebagai bagian untuk rekomendasi perbaikan sistem tersebut

Kata kunci— RFID, arsitektur sistem, ATAM, *reliability*, *performance*, *usability*.

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi identifikasi saat ini sudah semakin pesat. Salah satunya adalah RFID (*Radio Frequency Identification*). RFID adalah teknik identifikasi berbasis gelombang radio untuk mengidentifikasi barang ataupun makhluk hidup tertentu secara real-time dengan menggunakan Transponder [1]. Teknologi RFID saat ini banyak digunakan di berbagai bidang kehidupan manusia [13]. Setiap sistem memiliki arsitektur [10], begitu juga dengan teknologi RFID. Arsitektur teknologi ini disesuaikan dengan sistem yang menerapkan teknologi ini. *Software architecture* merupakan kumpulan struktur yang diperlukan untuk menjelaskan sistem, yang terdiri dari elemen perangkat lunak, hubungan tiap elemen, dan sifat keduanya [10]. *Software architecture* mencakup keputusan arsitektur yang berisi respon beberapa persyaratan atribut kualitas. Untuk menilai keterpenuhan persyaratan atribut kualitas, evaluasi *software architecture* perlu dilakukan pada tiap tahap pengembangan perangkat lunak.

Telkom University telah menerapkan teknologi RFID sebagai penunjang kegiatan perkuliahan. Teknologi RFID diterapkan pada sistem absensi mahasiswa dan dosen masing-masing mata kuliah. Dalam penerapannya terdapat beberapa kendala pada atribut kualitas sistem, antara lain kurang jelasnya notifikasi untuk mahasiswa atau dosen apakah absensi diterima atau tidak, waktu pembacaan kartu yang kadang cukup lama, kartu yang tidak terbaca oleh reader dan lain-lain. Untuk itu perlu dilakukan evaluasi

kualitas arsitektur sistem absensi agar sistem menjadi lebih baik.

Banyak metode yang dapat dilakukan untuk melakukan evaluasi *software architecture* [9][7]. Salah satu metodenya adalah *Architecture Tradeoff Analysis Method* (ATAM). ATAM merupakan metode berbasis scenario untuk melakukan analisis timbal balik dan evaluasi *software architecture* terhadap beberapa atribut kualitas perangkat lunak [12]. Tujuan dari metode ini adalah untuk identifikasi resiko dan non resiko serta hubungan timbal balik tiap masing-masing resiko dari arsitektur sistem.

Dengan menggunakan metode ATAM, penelitian ini akan melakukan evaluasi terhadap arsitektur sistem RFID presensi online pada Telkom University. Hasil yang diharapkan adalah evaluasi kualitas arsitektur sistem RFID dan rekomendasi perbaikan sistem berdasarkan hasil evaluasi tiap atribut kualitas terkait.

2. LANDASAN TEORI DAN METODE PENELITIAN

2.1 Radio Frequency Identification

RFID (*Radio Frequency Identification*) adalah merupakan sistem berbasis gelombang radio dengan frekuensi berbeda untuk melakukan identifikasi manusia, barang, makhluk lain atau apapun secara otomatis dengan menggunakan alat yang biasa disebut dengan RFID Tag atau Transponder. Teknologi RFID tergantung pada komunikasi antara RFID Tag dan RFID Reader. Tidak seperti teknologi *barcode*, RFID memberikan kemampuan untuk membaca,

menulis, transmisi, dan update informasi secara cepat. Kapasitas penyimpanan data pada *barcode* yang terbatas dan tidak ada kemampuan untuk di program ulang menyebabkan munculnya teknologi RFID yang bisa menyimpan data pada suatu silikon chip. Teknologi RFID sudah menjadi salah satu dari sepuluh teknologi penyumbang terbesar pada abad ke 21[13].

Secara garis besar, terdapat 3 komponen utama yang membangun sistem RFID [1], yaitu

1. RFID Tag, berisi chip semikonduktor yang sangat kecil dan antena. Chip tersebut menyimpan kode atau serial number yang unik untuk masing-masing objek tergantung tipe memori.
2. RFID Reader, merupakan alat pemindai yang memungkinkan membaca tag dan berkomunikasi dengan host computer.
3. RFID Database, merupakan tempat penyimpanan data atau informasi tentang objek yang akan diidentifikasi pada chip RFID.

2.2 Software Architecture

Software Architecture atau arsitektur perangkat lunak merupakan sekumpulan struktur yang diperlukan untuk menjelaskan tentang sistem, yang terdiri dari elemen perangkat lunak, hubungan antar elemen, dan sifat-sifat keduanya [9]. Setiap sistem memiliki *software architecture*, tetapi tidak semua sistem mendokumentasikan dan menyebarkan arsitekturnya tersebut.

Arsitektur perangkat lunak sangat penting bagi sebuah sistem. Hal ini dikarenakan arsitektur sistem dapat menghambat atau mengaktifkan atribut kualitas dari sistem. Beberapa jenis keputusan arsitektur menyebabkan atribut kualitas tertentu dalam sistem, sehingga keputusan arsitektur bisa dibuat terkait dengan atribut kualitas tersebut. Perbaikan sistem dapat segera dilakukan apabila kualitas sistem sudah diketahui dari arsitektur perangkat lunak.

2.3 Quality Attribute

Dalam sebuah sistem, terdapat kategori persyaratan dalam sebuah sistem. Salah satunya adalah persyaratan atribut kualitas sistem. Persyaratan ini memenuhi persyaratan dari persyaratan fungsional. Persyaratan fungsional merupakan kemampuan sistem untuk melakukan fungsionalitas sesuai dengan kriteria yang ditentukan. Dalam atribut kualitas, satu fungsionalitas sistem dapat membuat beberapa pertimbangan kualitas, seperti *performance*, *availability*, *reliability*, dan lain-lain. Dari pertimbangan tersebut digunakan skenario atribut kualitas sebagai karakterisasi atribut kualitas.

Skenario atribut kualitas dapat direpresentasikan menjadi 6 bagian [10], yaitu:

1. *Source of stimulus* merupakan beberapa entitas yang diperlukan oleh stimulus
2. *Stimulus* merupakan suatu kondisi yang membutuhkan response saat berada di sistem
3. *Environment* merupakan kondisi terjadinya stimulus, seperti kelebihan beban atau kondisi normal
4. *Artifact* merupakan rangsangan dari semua sistem atau beberapa sistem
5. *Response* merupakan aktivitas yang dilakukan sebagai hasil dari kedatangan stimulus
6. *Response measure* merupakan pengukuran dari respon yang terjadi melalui beberapa mode

2.4 Software Architecture Evaluation

Evaluasi arsitektur perangkat lunak menjadi salah satu teknik penting dalam penjaminan mutu perangkat lunak. Prinsip dalam evaluasi arsitektur perangkat lunak adalah untuk menilai potensi arsitektur yang digunakan untuk mewujudkan sistem yang mampu memenuhi persyaratan atribut kualitas yang dibutuhkan. Evaluasi perangkat lunak merupakan cara yang efektif untuk memahami hubungan timbal balik desain perangkat lunak dan dapat menentukan karakteristik atribut kualitas desain arsitektur perangkat lunak.

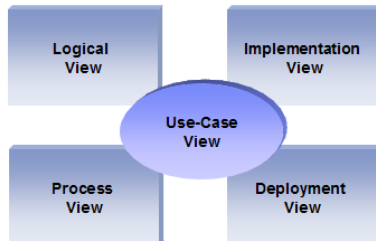
Evaluasi arsitektur dapat digunakan sebagai proses pengambilan keputusan. Proses pengambilan keputusan terdiri dari identifikasi masalah, analisis masalah dan pembuatan solusi, seleksi dan evaluasi. Meskipun evaluasi ini fokus pada seleksi dan evaluasi, tapi hal ini sering mencakup pembuatan solusi pada proses yang berulang.

2.5 Software Architecture Document

Keputusan desain arsitektural yang didapat dari beberapa *stakeholder* dapat didokumentasikan dengan jelas agar *stakeholder* dapat memahami maksud dari keputusan arsitektural. Setiap *stakeholder* memiliki kebutuhan yang berbeda-beda. Sehingga dalam pembuatan dokumen arsitektur terdapat gambar arsitektural yang berbeda-beda yang disebut dengan *architecture view*.

2.6 4+1 Architectural View Model

4+1 *Architectural View Model* merupakan model view untuk mendeskripsikan arsitektur sistem berdasarkan penggunaan multiple view. Model ini didesain oleh Philippe Kruchten pada tahun 1995[5]. Multiple view ini digunakan untuk memisahkan kepentingan yang berbeda-beda dari masing-masing *stakeholder* sistem dan memisahkan kebutuhan fungsional dan non fungsional sistem.



Gambar 2.1 4+1 Architecture View Model [11]

a. **Logical View**

- **Audience:** Designers
- **Area:** Functional Requirements: Mendefinisikan desain model objek, use case realizations dan business requirements dari sistem
- **Related Artefact:** Design model

b. **Process View**

- **Audience:** Integrators
- **Area:** Non-functional Requirements: Mendefinisikan aspek konkurensi dan sinkronisasi desain sistem
- **Related Artefact:** tidak ada artefact tertentu

c. **Deployment View**

- **Audience:** Manager Deployment
- **Area:** Topologi: menggambarkan pemetaan perangkat lunak ke perangkat keras dan menunjukan aspek distribusi sistem
- **Related Artefact:** Model Deployment

d. **Implementation View**

- **Audience:** Programmer
- **Area:** Komponen Sistem: Menggambarkan modul dan subsistem dari aplikasi
- **Related Artefact:** Model Implementasi

e. **Use-Case View**

- **Audience:** Semua stakeholder sistem termasuk end user
- **Area:** Menggambarkan set scenario use case yang mewakili beberapa fungsi utama yang signifikan dari sistem. Menjelaskan actor dan use case untuk sistem, view ini menyajikan kebutuhan user dan akan diuraikan lebih lanjut di tingkat desain secara lebih rinci
- **Related Artefact:** Use Case Model

2.7 Architecture Tradeoff Analysis Method

ATAM merupakan metode evaluasi berbasis skenario. Skenario dibuat melalui pertemuan dengan *stakeholder* dan analisis kebutuhan sistem. Ketika terdapat ketidaksesuaian antara sudut pandang *stakeholder* yang berbeda, dilakukan negosiasi untuk mendapatkan hasil akhir. Skenario yang didapat digunakan untuk menilai dukungan arsitektur dengan atribut kualitas sehingga dapat diidentifikasi elemen arsitektur yang mempengaruhi atribut kualitas. Skenario ini digunakan untuk memahami resiko pada arsitektur, non-resiko, poin sensitivitas terhadap atribut kualitas, dan timbal balik antara atribut kualitas.

Secara umum, terdapat 9 langkah yang dapat dilakukan dengan ATAM, dimana 9 langkah ini dikategorikan menjadi 4 kelompok konseptual [9], yaitu:

1. Presentasi, pertukaran informasi tentang sistem dengan presentasi dengan *stakeholder*. Terdapat 3 langkah pada presentasi:

- a. *Present The ATAM*. Menjelaskan tentang ATAM kepada para *stakeholder* mulai dari tahapan dalam melakukan evaluasi hingga hasil dari evaluasi arsitektur.
- b. *Present business driver*. Menggambarkan *functional requirement*, *business goal* dan dokumen arsitektural dari sistem
- c. *Present architecture*. Menjelaskan tentang arsitektur sistem berdasarkan model 4+1 view pada dokumentasi arsitektural sistem

2. Investigasi dan Analisis, menilai kebutuhan atribut kunci kualitas dan pendekatan arsitektur. Dalam kelompok ini terdapat 3 langkah:

- a. *Identify architectural approaches.*
Menidentifikasi pendekatan arsitektur tapi tidak menganalisisnya
- b. *Generate quality attribute utility tree.* Faktor kualitas yang membentuk sistem ditentukan hingga tahap skenario, dijelaskan dengan stimulus dan respon, dan diprioritaskan
- c. *Analyze architectural approaches.* Melakukan analisis pendekatan arsitektur yang membahas faktor-faktor dengan prioritas tinggi pada langkah 5. Melakukan identifikasi resiko arsitektur, poin sensitivitas dan poin timbal balik.

3. Pengujian, memeriksa hasil yang didapat dengan kebutuhan *stakeholder* terkait. Pada pengujian terdapat 2 langkah, yaitu:

- a. *Brainstorm and prioritize scenario.* Kumpulan skenario besar ditimbulkan dari *stakeholder* dan diprioritaskan melalui proses pemilihan
- b. *Analyze architectural approaches.* Skenario dengan peringkat tertinggi digunakan sebagai kasus pengujian. Skenario dimasukkan kedalam pendekatan arsitektur pada langkah 4. Pendekatan lain, resiko, poin sensitivitas dan poin timbal balik dapat diidentifikasi.

4. Laporan, melaporkan hasil akhir dari ATAM kepada *stakeholder*. Terdapat 1 langkah pada kelompok laporan:

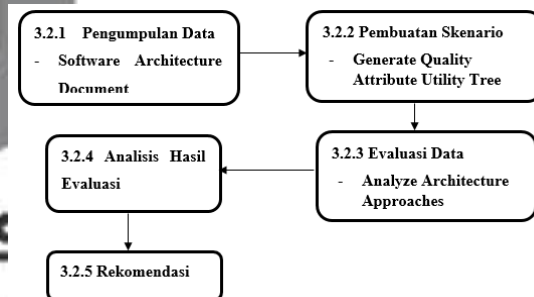
- c. *Present result.* Dari hasil yang didapat, (pendekatan arsitektur,

skenario, atribut kualitas, *utility tree*, resiko, non resiko, poin sensitivitas dan timbal balik) akan dipresentasikan ke semua *stakeholder*.

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Umum

Pada penelitian ini, penulis akan melakukan analisis terhadap hasil evaluasi dan sensitivitas Sistem RFID Universitas Telkom dengan menggunakan ATAM. Penggunaan ATAM lebih lanjut dilakukan pada bagian pengujian dan analisis. Pada analisis ini dilakukan dengan beberapa tahap, yaitu: pengumpulan data, pembuatan skenario, evaluasi data, analisis hasil evaluasi, dan rekomendasi. Dalam analisis ini tidak semua langkah dalam ATAM dilakukan, hal ini dikarenakan perlu adanya team evaluasi yang banyak untuk melakukan evaluasi dengan ATAM. Sehingga dalam tugas akhir ini hanya sebagian dari langkah ATAM yang dilakukan.



Gambar 3-1 Tahapan Proses

Penjelasan mengenai gambaran umum yang terdapat pada gambar 3-1, aktifitas yang dilakukan adalah sebagai berikut:

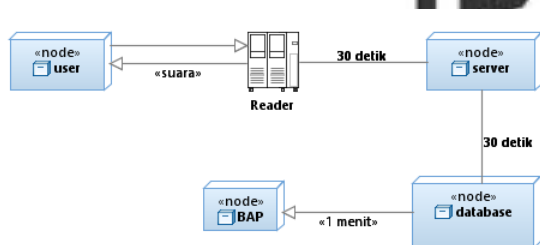
1. Data yang digunakan pada tugas akhir ini adalah *Software Architecture Document* Sistem RFID Universitas Telkom yang diperoleh dari Direktorat Sistem Informasi Telkom University. Dalam dokumen ini, penggambaran sistem RFID Universitas Telkom digambarkan dengan konsep '4+1' model view. Architectural view yang digambarkan pada dokumen ini adalah:
 - *Logical view*
 - *Process view*
 - *Deployment view*
 - *Implementation view*
 - *Use Case view*

2. Dari data yang didapatkan, dibuat pendekatan arsitektur sistem untuk menentukan faktor kualitas yang akan dilakukan evaluasi. Dari arsitektur sistem yang ada dapat ditentukan utility tree untuk faktor kualitas sistem. Faktor kualitas yang akan dilakukan analisis adalah reliability, performance, dan usability. Dari masing-masing faktor kualitas tersebut kemudian dibuat skenario sebagai acuan untuk melakukan analisis dan evaluasi arsitektur sistem. Tiap skenario kualitas sistem terdiri dari source, stimulus, artifact, environment, response dan response measure.
3. Dari skenario yang dibuat akan dibandingkan dengan arsitektur sistem, apakah arsitektur tersebut sudah dapat menangani permasalahan kualitas sistem.
4. Setelah dilakukan evaluasi sistem, hasil evaluasi akan dianalisis dengan arsitektur sistem yang sudah dibuat. Analisis ini dilakukan untuk mengetahui poin tradeoff, resiko dan non resiko dari pendekatan arsitektur sistem
5. Dari hasil analisis evaluasi arsitektur sistem, didapat keputusan desain arsitektural untuk perbaikan sistem RFID universitas Telkom. Keputusan desain arsitektural bisa merupakan bentuk arsitektur baru untuk sistem ataupun perbaikan dari arsitektur sistem yang telah ada.

3.2 Analisis Tradeoff

Dari arsitektural sistem yang sudah ada, diberikan penggambaran arsitektur alternative untuk perbaikan sistem. Arsitektur alternative ini kemudian dilakukan analisis terhadap skenario dari masing-masing atribut kualitas.

a. Alternative 1



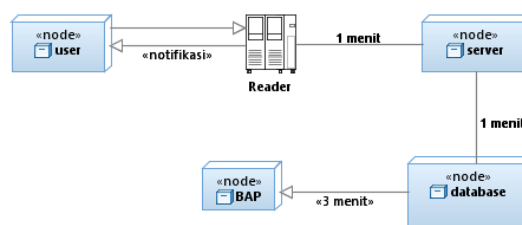
Gambar 3-2 Alternative Desain 1

Pada alternative desain yang pertama, notifikasi pada reader tetap menggunakan bunyi 1 kali atau 2 kali dan waktu pembacaan data dari reader ke database tetap sekitar 1 menit. Perbaikan dilakukan hanya pada waktu pembuatan BAP dari database pada igracias. Waktu untuk munculnya BAP pada igracias dibuat lebih cepat yakni selama 1 menit.

Desain arsitektur ini akan membuat waktu munculnya BAP pada igracias dosen menjadi lebih cepat

meskipun dosen baru melakukan tapping setelah semua mahasiswa melakukan tapping. Akan tetapi notifikasi yang diberikan reader hanya bunyi 1 kali atau 2 kali, sehingga mahasiswa harus melakukan pengecekan kembali pada akun igracias masing-masing apakah presensi mereka diterima atau tidak. Selain itu waktu pembacaan data dari reader hingga ke database tetap sekitar 1 menit, sehingga jika waktu pembacaan data telah habis maka reader akan mengeluarkan bunyi 1 kali yang berarti data ditolak padahal presensi mahasiswa tersebut diterima.

b. Alternative 2



Gambar 3-3 Alternative Desain 2

Pada alternative yang kedua, perbaikan dilakukan pada reader yang memberikan notifikasi berupa tulisan apakah presensi diterima atau tidak dan waktu pembacaan data pada reader hingga database menjadi lebih lama sekitar 2 menit. Tapi waktu untuk munculnya BAP pada igracias dosen tetap sekitar maksimal 3 menit.

Dengan desain ini, mahasiswa tidak perlu melakukan pengecekan kembali ke akun igracias masing-masing karena sudah ada notifikasi pada reader bahwa presensi mahasiswa diterima atau tidak. Selain itu waktu pembacaan data dari reader hingga ke database menjadi lebih lama sekitar 2 menit agar reader tidak mengeluarkan notifikasi ambigu apakah presensi diterima atau tidak. Tapi waktu munculnya BAP pada igracias dosen apabila dosen melakukan tapping setelah semua mahasiswa melakukan tapping tetap lama sekitar 3 menit, karena dibutuhkan waktu untuk melakukan sinkronisasi data sesuai ruangan dan jadwal mata kuliah dosen tersebut.

3.3 Rekomendasi

Dari hasil analisis tradeoff kedua arsitektur alternative, maka dipilih arsitektur alternative kedua untuk dijadikan perbaikan arsitektur sistem RFID pada universitas Telkom. Hal ini dikarenakan pada arsitektur alternative kedua, mahasiswa tidak lagi harus melakukan pengecekan presensi lagi ke igracias karena reader sudah mengeluarkan notifikasi bahwa absensi mahasiswa tersebut diterima. Selain itu reader juga tidak akan memberikan notifikasi ambigu apakah data diterima atau tidak karena waktu pengecekan data dari reader hingga ke database lebih lama sekitar 2 menit. Dan waktu munculnya BAP selama 3 menit

merupakan waktu normal setelah dosen melakukan tapping setelah semua atau beberapa mahasiswa melakukan tapping.

Berdasarkan hasil tradeoff tersebut maka rekomendasi untuk perbaikan sistem adalah dengan menambahkan notifikasi seperti tulisan pada reader bahwa presensi mahasiswa diterima dan menambahkan waktu untuk pengecekan data pada reader hingga ke database menjadi sekitar 2 menit agar reader tidak memberikan notifikasi ambigu apakah presensi mahasiswa ditolak atau diterima.

4. KESIMPULAN

Berdasarkan hasil evaluasi yang telah dilakukan sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. *Architecture Tradeoff Analysis Method* dapat digunakan untuk melakukan analisis dan evaluasi terhadap arsitektur sistem RFID pada Universitas Telkom dengan menurunkan atribut kualitas lalu membuat skenario pendekatan arsitektur sistem
2. Rekomendasi untuk dilakukan perbaikan sistem RFID Universitas Telkom berdasarkan hasil evaluasi atribut kualitas adalah menambahkan notifikasi pada reader seperti keterangan bahwa absensi diterima agar mahasiswa dapat lebih yakin bahwa absensi diterima, waktu pada reader untuk pengecekan data hingga pengiriman data ke reader lebih lama sehingga tidak mengeluarkan bunyi ambigu apakah data diterima atau tidak

5. SARAN

1. Penggunaan atribut kualitas yang lain seperti *security*, *portability* agar dapat mengetahui lebih lanjut tentang arsitektur sistem RFID
2. Penggunaan metode evaluasi arsitektur sistem yang lain agar dapat mengetahui perbandingan tiap metode untuk melakukan evaluasi arsitektur sistem

DAFTAR PUSTAKA

- [1] Ahsan, K., Shah, H., & Kingston, P. (2010). RFID Applications: An Introductory and Exploratory Study. *IJCSI International Journal of Computer Science Issues* Vol. 7.
- [2] Anonim. (2014, April 8). *Architecture Tradeoff Analysis Method*. Retrieved from <https://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm>
- [3] Babar, M. A., & Gorton, I. (n.d.). Comparison of Scenario-Based Software Architecture Evaluation Methods.
- [4] Chao, C.-C., Yang, J.-M., & Jen, W.-Y. (2007). Determining technology trends and forecasts of RFID by a historical review and bibliometric analysis from 1991 to 2005. *Technovation* 27.
- [5] Choi, H., & Yeom, K. (2002). An Approach to Software Architecture Evaluation with the 4+1 View Model of Architecture. *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. IEEE.
- [6] Clements, P., Bergey, J., & Mason, D. (2005). Using the SEI Architecture Tradeoff Analysis Method to Evaluate WIN-T: A Case Study. *Software Architecture Technology Initiative*.
- [7] Dobrica, L., & Niemela, E. (2002). A Survey on Software Architecture Analysis Methods. *IEEE Transactions on Software Engineering*, Vol. 28, No. 7.
- [8] Domdouzis, K., Kumar, B., & Anumba, C. (2007). Radio-Frequency Identification (RFID) applications: A brief introduction. *Advanced Engineering Informatics* 21.
- [9] Kazman, R., Bass, L., Lattanze, T., & Northrop, L. (2005). A Basis for Analyzing Software Architecture Analysis Method. *Software Quality Journal*, 329-355.
- [10] Kazman, R., Clements, P., & Bass, L. (2012). *Software Architecture In Practice Third Edition*. Addison-Wesley Professional.
- [11] Maréchaux, J.-L. (2005). Developing a J2EE Architecture with Rational Software Architect Using the Rational Unified Process.
- [12] Smith, D., & Merson, P. (2003). Using Architecture Evaluation to Prepare a Large Web Based System for Evolution. *Proceedings of the Fifth IEEE International Workshop on Web Site Evolution*. IEEE.

- [13] Sun, C., & Jiang, F. (2013). Research on RFID Applications in Construction Industry. *Journal of Networks*, Vol. 8, No. 5.
- [14] Zhu, L., Aurum, A., Gorton, I., & Jeffery, R. (2005). Tradeoff and Sensitivity Analysis in Software Architecture Evaluation Using Analytic Hierarchy Process. *Software Quality Journal*, 13.

