

## Self-Adaptive Systems : Fondasi untuk Pengembangan Smart City

Iping Supriana<sup>1)</sup>, Kridanto Surendro<sup>2)</sup>, Aradea<sup>3)</sup>

<sup>1,2</sup> Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung

<sup>3</sup> Teknik Informatika Fakultas Teknik Universitas Siliwangi Tasikmalaya

e-mail: [iping@informatika.org](mailto:iping@informatika.org); [endo@informatika.org](mailto:endo@informatika.org); [aradea.informatika@gmail.com](mailto:aradea.informatika@gmail.com)

### Abstrak

Karakteristik sebuah kota mencakup unsur sosial dan teknis yang sangat beragam, hal ini disebabkan oleh dinamika perubahan dan pertumbuhan lingkungannya yang sangat cepat. Dengan demikian suatu sistem yang mengaturnya dituntut memiliki kemampuan untuk beradaptasi. Saat ini telah lahir suatu konsep yang dikenal sebagai sistem smart city. Namun dari berbagai solusi yang ada terkadang terlalu menekankan pada faktor sosial tanpa memperhatikan model teknis secara matang, atau sebaliknya, sangat mengedepankan kemampuan teknis tanpa mempertimbangkan domain persoalan sebagai representasi dunia nyata. Makalah ini mengusulkan model self-adaptive system yang dijadikan sebagai landasan untuk mengembangkan smart city. Model dipersiapkan melalui dua strategi, yaitu kemampuan dalam memahami domain persoalan melalui pendekatan berbasis goal, dan kemampuan teknis dalam mengatur adaptasi komponen melalui pendekatan strategi kontrol. Berdasarkan hasil pembahasan, model yang diusulkan telah mampu merepresentasikan kebutuhan sistem smart city secara menyeluruh berdasarkan pedoman pengembangan yang direkomendasikan.

**Kata kunci:** self-adaptive systems, smart city, goal-oriented requirements engineering, control strategy

### 1. Pendahuluan

Suatu kota memiliki ruang kompleksitas persoalan yang sangat luas, hal ini berhubungan dengan isu-isu terkait bidang sosial, ekonomi, dan lingkungannya yang menjadi saling terhubung satu sama lain [1] melalui konsep teknis yang dikembangkannya. Kondisi ini menuntut suatu kota harus memiliki kemampuan dalam menciptakan sistem yang cerdas, dimana sistem dapat mengelola beragam sumber dayanya. Dimensi yang berhubungan dengan kebutuhan ini dapat terdiri dari kriteria kebutuhan untuk tingkatan kompetitif, mobilitas akses dan konektivitas, keberlanjutan sumberdaya alam, pengembangan human capital, tingkatan kualitas hidup, dan partisipasi pemerintah [2]. Deskripsi ini mencerminkan bahwa karakteristik suatu kota sangat beragam yang melingkupi unsur sosial dan teknis atau dikenal dengan *socio-technical system*. Selain itu, kondisi lingkungan yang rentan terhadap perubahan dan pertumbuhannya menjadi persoalan tambahan yang hadir didalam kompleksitas suatu kota.

Saat ini telah lahir suatu konsep yang dapat menjadi solusi atas persoalan tersebut dengan menawarkan berbagai kemudahan dalam mengelola sistem kota, istilah yang digunakan pun beragam sebut saja misalnya *cyber city*, *smart city*, *intelligent city*, dan lain-lain. Namun dari berbagai istilah tersebut perlu ditekankan bahwa kemampuan dalam menangkap dan memahami karakteristik, serta mengatur kemampuan responsivitas atas perubahan dan pertumbuhannya secara menyeluruh merupakan faktor utama penentu keberhasilan konsep tersebut. Terkadang dari beberapa solusi yang ada konsep terlalu menekankan pada persoalan sosial yang berkonsentrasi pada pemodelan konseptual tanpa memperhatikan persoalan teknis secara matang, atau sebaliknya, konsep sangat mengeksploitasi kemampuan teknis tanpa mempertimbangkan domain persoalan secara rinci sebagai representasi dunia nyata suatu lingkungan kota.

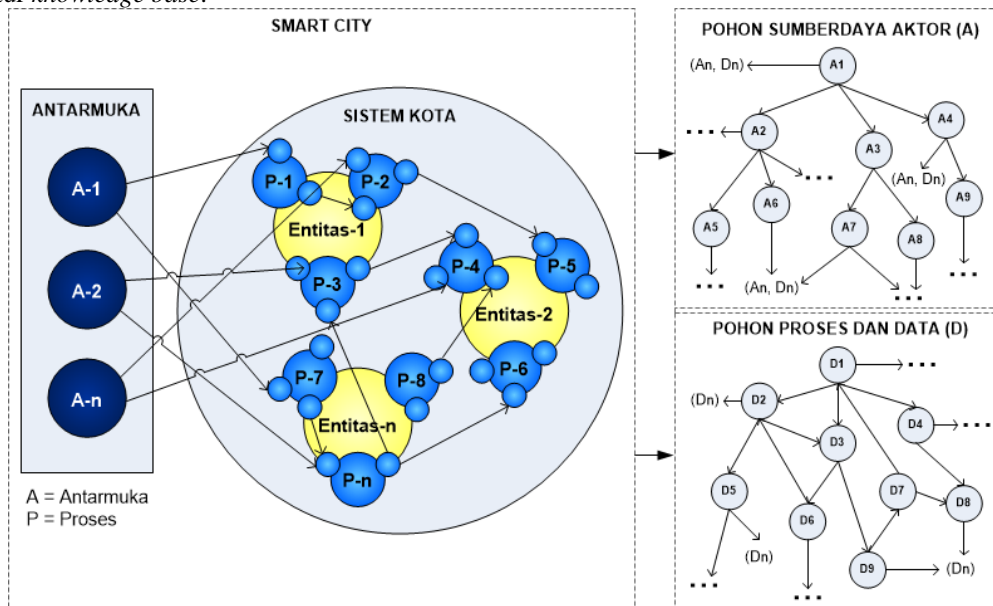
Makalah ini mengusulkan model *smart city* yang dilandasi oleh konstruksi dasar *self-adaptive systems* (SAS) [3]. Model terdiri dari dua strategi, pertama memodelkan sistem kompleks seperti sebuah kota melalui pendekatan *goal-oriented requirements engineering* untuk menangkap dan memahami domain persoalan sosial, dan kedua mengatur adaptasi komponen sistem secara teknis melalui penyediaan struktur pengetahuan dan mesin inferensi dengan pendekatan *control strategy*. Bagian ke-2 makalah ini mendeskripsikan model yang diusulkan, bagian 3 menguraikan pembahasan studi kasus dan pedoman pengembangannya, terakhir bagian 4 menyimpulkan hasil pembahasan dan pekerjaan masa depan.

## 2. Metodologi

Langkah-langkah pengembangan model yang diusulkan dilandasi atas prinsip SAS [3], yaitu merupakan kemampuan sistem untuk menyesuaikan perilakunya dalam menanggapi perubahan baik didalam sistem itu sendiri maupun lingkungannya [4][5]. Saat ini kemampuan dari SAS banyak digunakan para peneliti untuk berbagai kebutuhan dalam menyelesaikan ragam persoalan pada bidang yang berbeda [6]. Salah satu konsep dasar yang dapat digunakan adalah dengan mengintegrasikan kemampuan SAS kedalam aktivitas pemodelan requirements yang mendeskripsikan kondisi dunia nyata [7]. Dalam penelitian ini kami memperluas mekanisme tersebut untuk kebutuhan sistem *smart city*. Berdasarkan prinsip-prinsip SAS, terdapat dua aktivitas model yang dikembangkan, terdiri dari :

- Model domain sebagai domain persoalan dunia nyata yang dibentuk melalui pendekatan *goal-oriented requirements engineering*, model ini difungsikan untuk mendeskripsikan kebutuhan dunia nyata yang direpresentasikan sebagai sistem berbasis agen sosial. Pembahasan secara rinci mengenai pemilihan kerangka kerja berorientasi agen ini dapat dilihat dalam makalah kami sebelumnya [8].
- Model kontrol sebagai komponen teknis yang mengontrol atau mengelola sistem target, model ini menerapkan pola aksi agen melalui *transition rules* sehingga merepresentasikan suatu strategi kontrol untuk kebutuhan *self-adaptation*. Mekanisme pengembangan struktur pengetahuan dan mesin inferensi dibahas secara rinci pada makalah kami sebelumnya [3].

Gambaran umum dari model yang diusulkan seperti dapat dilihat pada Gambar 1. Aktivitas sistem diawali melalui pemodelan domain (Gambar 1 kiri), dengan mengidentifikasi setiap entitas sistem, sehingga dapat terdefinisi proses-proses ( $P_n$ ) dan sumber dayanya, serta relasi satu sama lain, kemudian setiap proses tersebut dihubungkan pada setiap antar muka yang akan menjadi eksekutor sistem. Aktivitas ini dilakukan dengan pendekatan pemodelan berbasis goal. Langkah selanjutnya, setiap unsur sistem yang telah terdefinisi dipetakan kedalam komponen perangkat lunak (Gambar 1 kanan), untuk menentukan mekanisme operasional sistem, yaitu komponen-komponen yang diset secara dinamis atau statis melalui struktur *knowledge base*.

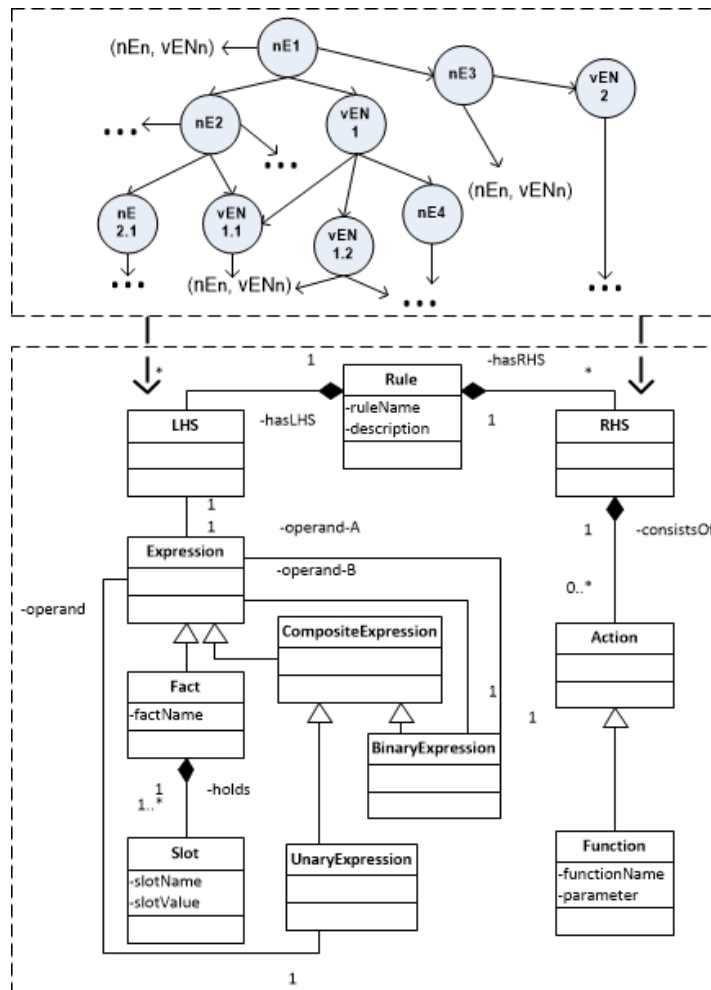


Gambar 1. Pemetaan model domain terhadap struktur pohon pengetahuan

Tahapan tersebut dilakukan untuk mengidentifikasi konteks dunia nyata dan menentukan mekanisme adaptasi, sehingga dari setiap konteks yang diidentifikasi dapat terbentuk struktur pohon pengetahuan, seperti ditunjukkan pada Gambar 1 kanan. Sebagai contoh, berdasarkan mekanisme identifikasi konteks anatarmuka pengguna, struktur pengetahuan dapat mengantisipasi pertumbuhan requirements pengguna, baik itu stakeholder internal maupun eksternal. Contoh lainnya, misalnya berdasarkan mekanisme identifikasi konteks layanan, struktur pengetahuan yang terbentuk dapat mengantisipasi perubahan dan pertumbuhan requirements aktivitas proses dan data, yang pada akhirnya menjadi layanan-layanan sistem. Selain itu, sangat dimungkinkan untuk menambahkan mekanisme identifikasi konteks lain untuk menentukan entitas-entitas sistem lainnya berdasarkan perubahan fakta atau pertumbuhan sistem.

Didalam struktur pohon pengetahuan tersebut setiap entitas sistem akan memiliki nama misalnya *slot nama-Entitas* ( $nE_1, nE_2, nE_3... nE_n$ ), dan memiliki *view* terhadap lingkungannya, misalnya *slot view-*

*Environment* ( $vEN$ ). Slot ini dapat terdiri dari fakta lainnya misalnya  $slot\ vEN_1, vEN_2, vEN_3, \dots vEN_n$ , dan setiap slot tersebut dimungkinkan memiliki spesifikasi lagi ( $vEN_{1.1}, vEN_{1.2}, \dots vEN_{1,n}$ ), dan sangat dimungkinkan spesifikasi tersebut dapat memiliki slot baru lainnya. Representasi fakta tersebut merupakan *list* pengetahuan yang dimiliki oleh setiap entitas, dan kami mendefinisikannya dalam bentuk suatu graf atau struktur pohon pengetahuan, seperti dapat dilihat pada Gambar 1 bagian kanan dan Gambar 2 bagian atas.



Gambar 2. Pohon pengetahuan dan sistem berbasis *rule* [3]

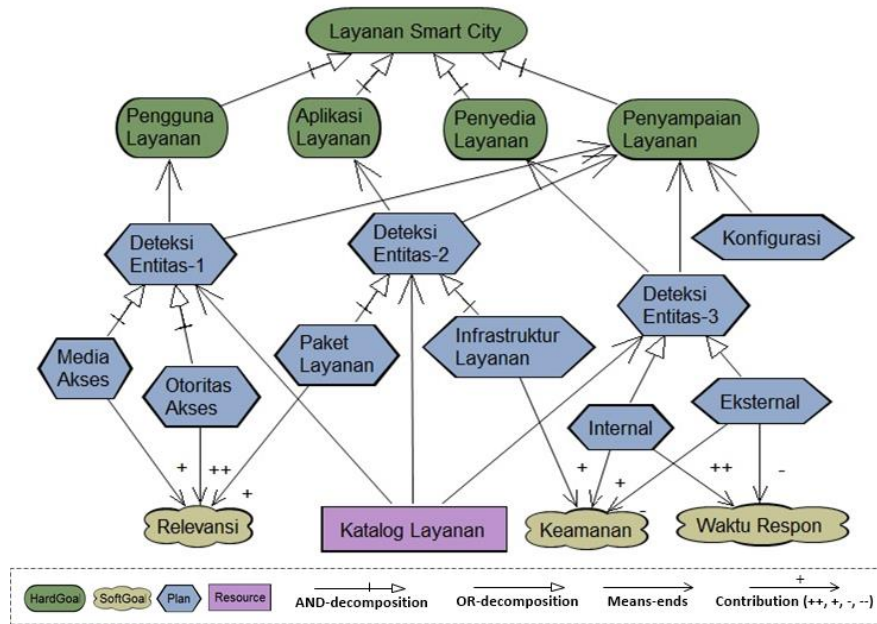
Struktur pohon pengetahuan ini akan secara otomatis menambahkan, menghapus, merubah, dan menyesuaikan setiap simpulnya, sebagai respon atas perubahan keadaan dari fakta-fakta yang terjadi. Didalam deskripsi struktur pohon pengetahuan tersebut, setiap propertinya dapat dikelola melalui model arsitektur agen yang merepresentasikan model domain dan pengaturan adaptasinya. Strategi yang dikembangkan, adalah melalui pembentukan *rule* untuk mengidentifikasi kesamaan properti dalam setiap *list* pengetahuan. Sistem berbasis *rule* umumnya terdiri dari 11 *classes* [9] yaitu, *Rule*, *left hand side (LHS)*, *Expression*, *Fact*, *Slot*, *Composite Expression*, *Unary Expression*, *Binary Expression*, *right hand side (RHS)*, *Action*, dan *Function*, seperti ditunjukkan pada Gambar 2 bagian bawah. Suatu aksi dalam RHS sangat ditentukan oleh ekspresi yang berada didalam LHS, ekspresi ini merupakan ekspresi tunggal (*pattern*) yang dikarakterisasi dengan nama dan koleksi *slot*, atau dapat menjadi ekspresi komposit dengan elemen kondisional (*and, or, not*), yang digunakan bersama untuk menghubungkan ekspresi tunggal (fakta) atau ekspresi komposit. Kelas fakta memiliki referensi *containment* kepada kelas *slot* dengan atribut *slotName* dan *slotValue*, dimana properti dari *slot* inilah yang dapat dideteksi perubahannya dari setiap unsur yang terkandung dalam setiap fakta domain model.

### 3. Hasil dan Pembahasan

Bagian ini mendeskripsikan penerapan model yang telah dibahas pada bagian 2 kedalam studi kasus layanan *smart city*, serta pedoman pengembangannya yang dapat digunakan oleh para pengembang untuk mewujudkan *self-adaptive smart city*.

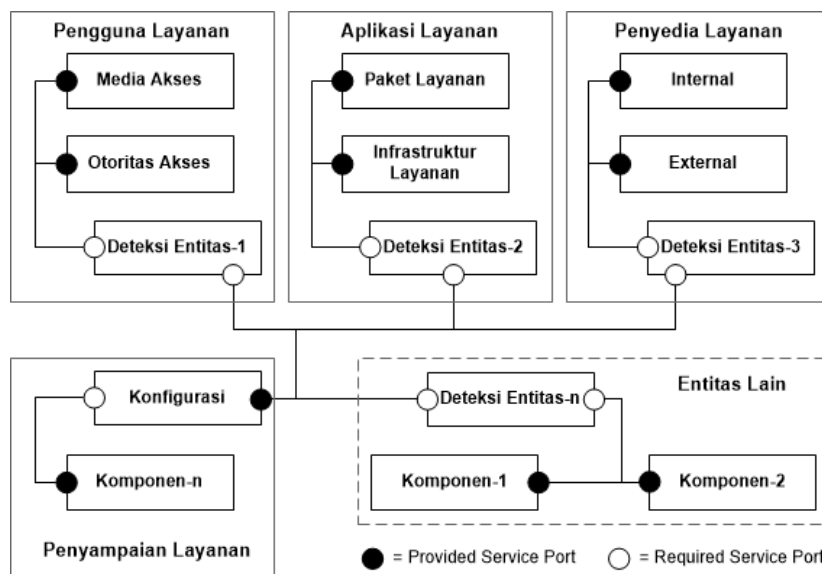
3.1. Studi Kasus

Aktivitas pengembangan diawali dengan pemodelan sistem seperti ditunjukkan pada Gambar 3, alat bantu yang kami gunakan adalah TAOM4E [10]. *Goal* utama sistem *smart city* adalah mewujudkan layanan *smart city* yang dikelola melalui 4 aktivitas *sub goal* yaitu pengguna, aplikasi, penyedia dan penyampaian layanan, dan setiap *sub goal* dapat memiliki *plan* yang berkontribusi terhadap *soft goal*.



Gambar 3. Pemodelan domain *smart city*

Tahapan selanjutnya model Gambar 3 dipetakan kedalam model komponen perangkat lunak [11][12] seperti disajikan pada Gambar 4. Masing-masing *goal* dan *plan* akan membentuk komponen primitif sehingga terdapat 4 kelompok komponen, 3 kelompok komponen berfungsi untuk mendeteksi entitas, ketiga kelompok komponen tersebut dikoordinasikan melalui komponen konfigurasi yang akan menentukan strategi adaptasi untuk setiap komponen lainnya. Artinya jika pada saat *run-time* dibutuhkan penambahan komponen baru, maka secara otomatis dapat ditambahkan seperti ditunjukkan didalam garis putus-putus pada Gambar 4. Begitu juga jika suatu komponen sudah tidak dibutuhkan atau perlu dirubah atau perlu digunakan kembali, maka dapat dieksekusi pada saat *run-time*.



Gambar 4. Spesifikasi komponen *smart city*

Berikut disajikan contoh spesifikasi komponen penyedia layanan pada Gambar 4 berdasarkan model komponen [11][12] :

```
Component LayananInternal {
  require : Konfigurasi;
  provide : LayananTI;
  mode : Dinamis;
  goal : PenyediaLayanan;
  softgoal : Keamanan+, WaktuRespon++;
  constraints: Set-2;
}

Component LayananEksternal {
  require : Konfigurasi;
  provide : LayananCloud;
  mode : Dinamis;
  goal : PenyediaLayanan;
  softgoal : Keamanan-, WaktuRespon-;
  constraints: Set-1;
}

Component DeteksiEntitas-3 {
  inst
  A : LayananInternal;
  B : LayananEksternal;
  bind
  serviceDelivery – A.konfigurasi;
  A.LayananTI – analyze;
  relation
  relation : OR(A,B);
}
```

Komponen layanan internal merupakan komponen primitif yang membutuhkan komponen konfigurasi, mode komponennya diinisialisasi oleh variabel dinamis karena dapat ditetapkan pada saat *run-time*. Pencapaian *goal* penyedia layanan menjadi target komponen ini yang diaktivasi berdasarkan kriteria *softgoal* dan *constraints*. *Softgoal* merupakan pertimbangan kontribusi terhadap sistem non fungsional, sementara *constraints* adalah asumsi domain yang ditetapkan. Komponen DeteksiEntitas-3 merupakan komponen komposit yang mendeskripsikan interkoneksi komponen primitif, serta kebutuhannya dengan komponen konfigurasi melalui tipe *OR-relation* pada pemodelan *goal*.

Berdasarkan konfigurasi komponen tersebut maka setiap komponen akan membentuk pengetahuan sesuai dengan kebutuhan fungsinya masing-masing. Sebagai contoh, *goal* penyedia layanan beserta *plan* deteksi entitas-3 akan membentuk pohon *service provider*, *goal* aplikasi layanan beserta *plan* deteksi entitas-2 akan membentuk pohon data atau layanan, serta *goal* pengguna layanan beserta *plan* deteksi entitas-1 akan membentuk pohon aktor atau pengguna, begitu juga dengan *goal* dan *plan* lainnya yang terbentuk pada saat *run-time*.

### 3.2. Pedoman Pengembangan

Berdasarkan seluruh pembahasan metodologi dan studi kasus, kami mendefinisikan langkah-langkah aktivitas yang dapat dijadikan sebagai pedoman bagi para pengembang dalam mengembangkan *self-adaptive systems* pada umumnya dan *self-adaptive smart city* pada khususnya, terdiri dari :

- Aktivitas diawali dengan pemodelan *goal* untuk merepresentasikan domain model (*problem domain*), pastikan setiap kebutuhan fungsional direpresentasikan sebagai *goal*, dan kebutuhan non-fungsional sebagai *soft-goal* yang dapat mencerminkan kriteria kualitas sistem.
- Tentukan parameter-parameter prioritas dan *constraints* untuk setiap kebutuhan fungsional dan kebutuhan non-fungsional tersebut.
- Memetakan deskripsi *goal* kedalam bahasa deskripsi arsitektur (misalnya ADL) untuk dikonfigurasi menjadi komponen perangkat lunak. Hal ini dapat dilakukan melalui generator perangkat lunak untuk membangkitkan kode komponen yang dibutuhkan.
- Seluruh komponen yang terbentuk dapat diatur melalui suatu kontrol untuk memberikan fungsi inferensi. Komponen ini menghasilkan spesifikasi model formal untuk melaksanakan tugas-tugas penalaran yang sesuai.
- Terakhir, suatu *inference engine* dapat dikembangkan dengan menyusun basis aturan sesuai dengan asumsi domain ataupun *policy* yang ditetapkan, hal ini harus dilakuan dengan memperhatikan karakteristik dinamis lingkungan dan preferensi dari setiap *stakeholder* yang terlibat.

## 4. Simpulan

Persoalan utama dalam mengembangkan sistem *smart city* berhubungan dengan unsur *socio-technical system* yang memiliki karakteristik keberagaman unsur dan sangat dinamis. Makalah ini mendeskripsikan model pengembangan *smart city* berlandaskan strategi *self-adaptive systems*. Aktivitas pengembangan sistem *smart city* diawali dengan pemodelan *goal* yang dapat mendukung pemodelan sosial dengan aktor, dependensi, dll. sehingga dapat merepresentasikan domain persoalan dari kondisi dunia nyata. Model *goal* ini dijadikan sebagai dasar untuk merancang perangkat lunak adaptif dengan memetakannya terhadap komponen sistem yang mendukung ruang perilaku adaptasi sebagai respon atas perubahan yang terjadi baik didalam sistem itu sendiri maupun lingkungannya. Kemampuan ini bisa digunakan saat *runtime* melalui mekanisme teknis dari strategi kontrol untuk mewujudkan penalaran terhadap seluruh komponen sistem berdasarkan pengetahuannya.

Makalah ini memberikan sudut pandang alternatif dalam mengembangkan sistem perangkat lunak adaptif pada umumnya dan sistem *smart city* pada khususnya, pendekatan yang diperkenalkan diharapkan dapat menjadi solusi bagi persoalan yang berhubungan dengan keberagaman unsur dan kedinamisan sifatnya. Sebagai pekerjaan kedepan kami berencana memperluas model ini dengan menyediakan kemampuannya dalam menangani ketidakpastian, dimana saat *runtime* sangat dimungkinkan pengetahuan dari sistem menjadi tidak lengkap atau tidak konsisten, hal ini dapat disebabkan adanya perubahan *requirements*, baik itu hilang atau ambigu, asumsi yang salah, entitas yang tidak terduga, atau fenomena lainnya didalam suatu lingkungan eksekusi.

#### **Daftar Pustaka**

- [1] ITU-T. Overview and role of ICT in smart sustainable cities. Telecommunication Standardization Sector, 2014.
- [2] Lombardi P. New challenges in the evaluation of smart cities. Network Industries Quarterly 13 (3): 8-10, 2011.
- [3] Supriana, I., Surendro, K., Aradea., Ramadhan, E., Self-adaptive cyber-city system. The Internet of Things: Foundation for Smart Cities, eHealth, and Ubiquitous Computing ISBN: 1351652095, 9781351652094. Pages 293-318. CRC Press Taylor & Francis Group, 2017.
- [4] Cheng, B.H.C., et. al. Software engineering for self-adaptive systems : a research roadmap, Self-Adaptive Systems, LNCS 5525, Springer-Verlag Berlin Heidelberg, 2009.
- [5] de Lemos, R., et. al., Software engineering for self-adaptive systems : a second research roadmap, Self-Adaptive Systems, LNCS 7475, pp. 1–32, Springer-Verlag Berlin Heidelberg, 2013.
- [6] Aradea., Supriana, I., Surendro, K., Darmawan, I. Variety of approaches in self-adaptation requirements: a case study. Recent Advances on Soft Computing and Data Mining, Advances in Intelligent Systems and Computing, Springer, Vol.549, pp.253-262, 2017.
- [7] Aradea., Supriana, I., Surendro, K., Darmawan, I. Integration of self-adaptation approach on requirements modeling. Recent Advances on Soft Computing and Data Mining, Advances in Intelligent Systems and Computing, Springer, Vol.549, pp.233-243, 2017.
- [8] Aradea., Supriana, I., Surendro, K. Prinsip paradigma agen dalam menjamin keberlangsungan hidup sistem. Prosiding Konferensi Nasional Sistem Informasi (KNSI 2015). ISSN : 1906-9613. Halaman 384. Universitas Klabat. Airmadidi Minahasa Utara, Sulawesi Utara, 2015.
- [9] Wu, C. G. Modeling rule-based systems with EMF. Eclipse Corner Article. Copyright © 2004 Chaur G. Wu. All rights reserved., 2004.
- [10] Tool for Agent-Oriented visual Modelling for Eclipse (TAOM4E) and its plugin t2x (Tropos4AS to Jadex), developed by the Software Engineering group at Fondazione Bruno Kessler (FBK), Trento, available, including the extensions, at <http://selab.fbk.eu/taom>.
- [11] Hirsch, D., Kramer, J., Magee, J., Uchitel, S. Modes for software architectures. In EWSA, pages 113–126. LNCS, 2006.
- [12] Horikoshi, H., Nakagawa, H., Tahara, Y., Ohsuga, A. Dynamic reconfiguration in self-adaptive systems considering nonfunctional properties. SAC'12 March 2630, Riva del Garda, Italy, ACM 2011.