

PERANCANGAN MODEL REPRESENTASI PENGETAHUAN BERBASIS ONTOLOGI PADA APLIKASI SIPELANTIK: STUDI KASUS PUSINTEK KEMENTERIAN KEUANGAN

Andrianto Susilo, Putu Wuri Handayani, dan Iik Wilarso

Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia
Jl. Salemba Raya No.4, Jakarta, 10430, Indonesia

E-mail: andrianto.susilo@ui.ac.id

Abstract

In supporting the implementation of ICT service management, Pusat Sistem Informasi dan Teknologi Keuangan (Pusintek) already have the Sistem Pengelolaan Layanan Teknologi Informasi dan Komunikasi (Sipelantik) application. This application is the automation of ICT service management business process, especially in the area of Service Support. In practice, the knowledge generated from this business process has broad enough coverage. However, these knowledges are not easily retrieved. Thus, a model that can describe and connect the knowledge is demanded. This study proposes a model of ontology-based ICT services knowledge representation using the existing knowledge on the Sipelantik application. The Ontology models are developed by software Protege in the form of Web Ontology Language (OWL). Simple Protocol And RDF Query Language (SPARQL) and Description Logic (DL) Query are used to obtain the information and knowledge from OWL models and to ensure that the relationships between concepts are properly defined. Reasoner Pellet plugin is used to ensure the consistency of the built ontology models. This research results in six classes of knowledge (knowledge base, role, service request, incident handling, component configuration, and service catalog) including the subclasses and an ontology model as knowledge of ICT services that can be shared and reused.

Keywords: *knowledge, knowledge representation, ontology*

Abstrak

Dalam mendukung pelaksanaan manajemen layanan TIK, Pusat Sistem Informasi dan Teknologi Keuangan (Pusintek) milik Kementerian Keuangan telah memiliki aplikasi Sistem Pengelolaan Layanan Teknologi Informasi dan Komunikasi (Sipelantik). Aplikasi Sipelantik merupakan otomatisasi dari proses bisnis manajemen layanan TIK khususnya pada area *Service Support*. Dalam praktiknya, pengetahuan yang dihasilkan pada proses bisnis ini memiliki cakupan yang cukup luas. Pengetahuan yang terdapat pada aplikasi Sipelantik tidak mudah untuk dicari atau dipelajari, sehingga dibutuhkan sebuah model yang dapat menggambarkan dan menghubungkan pengetahuan-pengetahuan tersebut. Penelitian ini mengusulkan model representasi pengetahuan layanan TIK berbasis ontologi berdasarkan pengetahuan pada aplikasi Sipelantik. Model ontologi yang dibangun menggunakan perangkat lunak *Protege* dalam bentuk *Web Ontology Language (OWL)*. *Simple Protocol And RDF Query Language (SPARQL)* dan *Description Logic (DL) Query* digunakan untuk mendapatkan informasi dan pengetahuan dari model OWL serta untuk memastikan hubungan antar konsep didefinisikan dengan benar. *Reasoner plugin* yaitu *Pellet* digunakan untuk memastikan konsistensi model ontologi yang dibangun. Penelitian ini menghasilkan enam kelas pengetahuan (basis pengetahuan, pengelola layanan, permintaan layanan, penanganan gangguan, komponen konfigurasi, dan katalog layanan) beserta subkelasnya dan model ontologi sebagai pengetahuan terkait layanan TIK yang dapat dibagikan atau digunakan kembali.

Kata kunci: *pengetahuan, representasi pengetahuan, ontologi*

1. Pendahuluan

Dalam rangka pelaksanaan manajemen layanan teknologi informasi dan komunikasi secara efektif dan optimal, Kemenkeu telah menetapkan suatu keputusan KMK Nomor 414/KMK.01/2011 Tentang Kebijakan dan Standar Manajemen Layanan Teknologi dan Komunikasi Area *Service Sup-*

port dan KMK Nomor 64/KMK.01/2012 Tentang Kebijakan dan Standar Manajemen Layanan Teknologi dan Komunikasi Area *Service Delivery*. Kebijakan tersebut berlaku untuk semua unit pengelola layanan TIK di lingkungan Kemenkeu. Pusintek (Pusat Sistem Informasi dan Teknologi Keuangan) sebagai unit pengelola layanan TIK (*shared service*) wajib untuk menerapkan kebijak-

an tersebut. Dari kebijakan tersebut, Pusintek telah membuat prosedur kerja pada proses-proses di kedua area.

Dalam mendukung pelaksanaan manajemen layanan TIK, Pusintek telah memiliki aplikasi Sistem Pengelolaan Layanan Teknologi Informasi dan Komunikasi (Sipelantik) sejak tahun 2008. Aplikasi Sipelantik merupakan otomatisasi dari proses bisnis manajemen layanan TIK khususnya pada area *Service Support*. Proses yang berjalan pada aplikasi tersebut meliputi Manajemen Gangguan Layanan TIK, Manajemen Masalah Layanan TIK, Manajemen Konfigurasi Layanan TIK, Manajemen Perubahan Layanan TIK, dan Manajemen Rilis Layanan TIK serta fungsi *Service Desk* sebagai *Single Point of Contact* untuk seluruh layanan TIK. Setiap proses dipimpin oleh satu orang Koordinator dan dibantu oleh beberapa Pelaksana Teknis.

Keberadaan aplikasi Sipelantik sangat penting untuk memudahkan pengguna atau Pelaksana Teknis dalam melakukan pengelolaan administrasi, distribusi, dan monitoring terhadap setiap tiket permintaan layanan TIK dan tiket pengaduan gangguan layanan TIK. Aplikasi ini merupakan sarana untuk mengontrol setiap layanan ataupun gangguan yang datang untuk selanjutnya diukur demi peningkatan kinerja layanan TIK Pusintek. Aplikasi Sipelantik merupakan sumber pengetahuan untuk pengambilan keputusan, analisis dampak, dan analisis resiko dalam mengelola layanan TIK di Pusintek. Namun, sumber pengetahuan dari proses bisnis yang terdapat pada aplikasi Sipelantik belum secara optimal dimanfaatkan oleh Pelaksana Teknis dalam menyelesaikan gangguan layanan terutamanya gangguan yang sama atau berulang. Pelaksana Teknis yang sedang menyelesaikan suatu gangguan layanan kesulitan mencari pengetahuan yang telah ada dan kesulitan mencari hubungan antar pengetahuan satu dengan yang lainnya sehingga pengetahuan yang ada belum optimal dapat digunakan untuk menyelesaikan gangguan.

Selain kendala dalam pemanfaatan basis pengetahuan, kendala lain juga terjadi pada komponen konfigurasi. Komponen konfigurasi saat ini masih dalam proses pemutakhiran data dan sebatas digunakan untuk menyimpan data. Aplikasi Sipelantik belum dapat digunakan untuk mengenali dan menampilkan hubungan antara komponen konfigurasi dengan komponen konfigurasi lainnya dan antara komponen konfigurasi ke layanan TIK. Hubungan antara komponen konfigurasi dengan komponen konfigurasi lainnya dan antara komponen konfigurasi ke layanan TIK dapat dijadikan sebagai salah satu sumber pengambilan keputusan dan analisis dampak dan resiko. Sebagai contoh bila sebuah *Server* mengalami masalah, *Service*

Desk dapat dengan cepat mengidentifikasi pengguna dan layanan mana yang berpotensi terkena dampak dari masalah *Server* tersebut.

Berdasarkan wawancara penulis dengan perwakilan dari manajemen Pusintek, implementasi Sipelantik akan maksimal apabila manajemen pengetahuan yang ada dalam sistem tersebut dikelola dengan baik, didukung dengan sistem yang optimal, dan terdapat keterkaitan pengetahuan yang satu dengan yang lainnya sehingga pengetahuan menjadi komprehensif. Pengetahuan yang terdapat pada aplikasi Sipelantik dapat menjadi komprehensif apabila dilakukan pengklasifikasian pengetahuan, penentuan hubungan antarpengertian dan pembangunan ontologi pengetahuan. Menurut Abburu & Babu cara efektif dan efisien dalam mengelola pengetahuan adalah dengan merepresentasikan pengetahuan [1]. Merepresentasikan pengetahuan berarti merepresentasikan kelas pengetahuan dan hubungan antar kelas pengetahuan [3]. Menurut Ji Ma, Xiping Pan, & Jiayi Yao ada dua cara klasik untuk merepresentasikan pengetahuan yaitu dengan jaringan semantik dan ontologi [7]. Ontologi memungkinkan untuk merepresentasikan suatu pengetahuan dan peran ontologi adalah untuk mendukung berbagi pengetahuan dan penggunaan kembali pengetahuan [9].

Berdasarkan fakta-fakta yang telah dijabarkan, maka dapat dirumuskan sebuah pertanyaan penelitian sebagai berikut: Bagaimana model ontologi yang tepat untuk merepresentasikan pengetahuan dalam domain layanan TIK Kementerian Keuangan pada aplikasi Sipelantik di Pusintek Kementerian Keuangan. Penelitian ini bertujuan untuk merancang model representasi pengetahuan berbasis ontologi pada domain layanan TIK pada aplikasi Sipelantik di Pusintek sehingga dapat digunakan untuk memudahkan berbagi dan penggunaan kembali pengetahuan pada domain layanan TIK.

2. Tinjauan Teoritis

Representasi Pengetahuan

Menurut Hua Wang representasi pengetahuan adalah suatu deskripsi atau spesifikasi dari pengetahuan [6]. Representasi pengetahuan berarti pengetahuan dapat diekspresikan sebagai simbol yang dapat diterima menggunakan komputer yang selanjutnya dapat dideskripsikan ke dalam beberapa bentuk seperti struktur grafik, struktur pohon, dan kaidah pola kecocokan [14]. Menurut Abburu & Babu representasi pengetahuan penting untuk efektivitas dan efisiensi domain manajemen pengetahuan [1]. Merepresentasi pengetahuan adalah mendefinisikan istilah dan kemudian mengkombinasikannya ke dalam suatu cara untuk menguraikan lebih banyak arti tentang area penge-

tahan. Merepresentasi pengetahuan adalah representasi *classes, instances, relationships, properties, dan rules* dari area pengetahuan [3].

Menurut Tanwar, P., Prasad, T. V., & Aswal, M. S. teknik representasi pengetahuan dapat dibagi menjadi tiga kategori yaitu logika predikat, jaringan semantik, dan bingkai [13]. Menurut Ji Ma, Xiping Pan, & Jiayi Yao ada dua cara klasik untuk merepresentasikan pengetahuan yaitu dengan jaringan semantik dan ontologi, yang keduanya dapat dideskripsikan dengan grafik [7]. Jaringan semantik lebih fleksibel untuk mendeskripsikan pengetahuan, sedangkan ontologi dapat mendeskripsikan pengetahuan lebih tepat dan mengelola pengetahuan dengan baik.

Ontologi

Ontologi diperkenalkan pertama kali oleh para filsafat. Ontologi adalah ilmu yang mempelajari segala sesuatu apa yang ada, dari jenis dan struktur objek, properti, peristiwa, proses dan hubungan disetiap bidang realitas [11]. Selain ontologi menjadi objek penelitian dalam filsafat, juga digunakan dalam penyelidikan, pengembangan, dan aplikasi dalam disiplin ilmu yang berhubungan dengan komputasi, informasi, dan pengetahuan (misalnya, kecerdasan buatan, representasi pengetahuan, perpustakaan, dan manajemen basis data) [8]. Menurut Haisheng Li, Wenzheng Li, Qiang Cai, & Hongzhi Liu, ontologi merupakan sejenis grafik khusus yang menggambarkan entitas domain, sifat, dan hubungan antara keduanya [5]. Ontologi memungkinkan untuk merepresentasikan suatu pengetahuan dan peran ontologi adalah untuk mendukung berbagi pengetahuan dan penggunaan kembali pengetahuan [9]. Ontologi merupakan sebuah teknologi yang efektif yang memungkinkan integrasi dengan sumber daya terkait, berbagi pengetahuan yang benar dan menghindari informasi yang tidak relevan dengan kata lain ontologi adalah teknologi yang baik untuk merepresentasikan pengetahuan suatu domain [1].

Ontologi menjadi populer dengan alasan sebagai berikut [11]:

- 1) *Reusability*; menciptakan ontologi baru dengan cara ontologi yang sudah ada dapat digunakan kembali.
- 2) *Formal community time*; berbagi sudut pandang yang luas terhadap wacana tertentu.
- 3) *Interoperability*; dukungan komunikasi dan kerjasama antarsistem.
- 4) *Knowledge level validation and verification*; memeriksa kelengkapan wacana domain.

Ontologi memiliki beberapa komponen yang dapat menjelaskan ontologi tersebut. Komponen utama dari ontologi menurut Gomez-Perez, A., & Benjamins, V.R. adalah *concept, relation, func-*

tion, axiom, dan instance [4]. Sebuah ontologi harus diekspresikan dalam notasi yang nyata. Sebuah ontologi mempunyai struktur bahasa yang formal. Bahasa ontologi memungkinkan pengguna untuk menulis secara eksplisit konseptualisasi formal dari model domain. Bahasa ontologi digunakan dalam representasi pengetahuan dalam hal ekspresi dan sifat komputasi. Beberapa komponen yang menjadi struktur bahasa ontologi antara lain *Extensible Markup Language (XML), XML Scheme, Resource Description Framework (RDF), RDF Scheme, dan Web Ontology Language (OWL)* [2].

Ontology Web Language (OWL) adalah bahasa ontologi yang direkomendasikan oleh W3C yang dikembangkan dari teknologi XML/S dan RDF/S. OWL merupakan lanjutan dari RDF Scheme. Menurut Horridge, OWL terdiri dari tiga komponen utama yaitu (a) individu yang merepresentasikan elemen pada suatu domain atau individu dapat dianggap sebagai anggota dari suatu kelas, (b) kelas yang menjelaskan konsep dalam suatu domain, dan (c) properti atau juga disebut *slot* bersifat relasi biner yang menghubungkan antara dua individu [15]. Terdapat beberapa *tool* yang mendukung bahasa ontologi bersifat *free, open source, client/Server, dan standalone*. Beberapa *tool* yang digunakan untuk mengembangkan ontologi diantaranya adalah Protégé, WebODE, OntoEdit, dan WebOnto.

Metodologi Pengembangan Ontologi

Terdapat banyak pendapat mengenai tahapan proses pengembangan model ontologi. Salah satunya menurut Noy, N.F., & McGuinness, D.L., tahapan yang dilakukan dalam proses pengembangan ontologi [10] adalah:

Menentukan domain dan ruang lingkup ontologi

Tahap ini merupakan tahap awal proses pengembangan ontologi. Pengembangan ontologi dimulai dengan mendefinisikan domain dan ruang lingkup dengan cara menjawab lima pertanyaan dasar yang dapat membantu untuk membuat daftar pertanyaan (*Competency Question*) yang dapat dijawab oleh model ontologi. Tiga dari lima pertanyaan dasar tersebut adalah:

- 1) Domain apa yang akan dikembangkan dalam ontologi?
- 2) Apa tujuan menggunakan ontologi?
- 3) Siapa yang akan menggunakan dan mengelola ontologi?

Mempertimbangkan penggunaan kembali ontologi yang sudah ada

Tahap ini melakukan pengecekan apakah ontologi yang sudah ada dapat digunakan kembali

atau perlu mengembangkan ontologi dari awal. Jika menggunakan ontologi yang sudah ada dapat dilakukan perbaikan dan perluasan.

Menentukan istilah penting dalam ontologi

Tahap ini menentukan daftar istilah yang berhubungan dengan ontologi yang dikembangkan. Daftar istilah dapat berupa istilah kelas dan properti dari kelas.

Mendefinisikan kelas dan hierarki kelas

Tahap ini mendefinisikan kelas ontologi dan menyusun kelas tersebut dalam hierarki taksonomi (*superclass-subclass*). Hierarki kelas, seperti tertera pada Tabel dan Gambar pada Lampiran, direpresentasikan dengan relasi bahwa jika kelas A adalah super kelas dari kelas B, setiap *instance* dari kelas B juga menjadi *instance* dari kelas A. Pendekatan dalam pengembangan hierarki kelas dapat menggunakan proses pengembangan *top-down* yang dimulai dengan mendefinisikan konsep yang umum dalam domain dilanjutkan dengan konsep yang lebih spesifik atau proses pengembangan *bottom-up* yang dimulai dengan mendefinisikan konsep yang paling spesifik kemudian dikelompokkan ke dalam konsep yang umum, dan proses pengembangan *combination* yang dimulai dengan mendefinisikan konsep yang penting kemudian menggeneralisasikan dan mengkhususkan konsep tersebut.

Mendefinisikan properti dari kelas - slot

Tahap ini mendefinisikan properti dari masing-masing kelas yang telah didefinisikan.

Mendefinisikan faset dari slot

Tahap ini mendefinisikan batasan dari setiap properti dari masing-masing kelas. Batasan yang dimaksud berupa *slot* kardinalitas dan tipe data.

Membuat instance

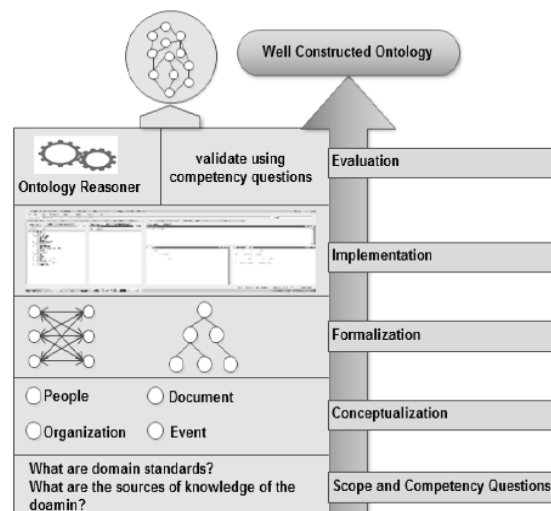
Tahap ini mendefinisikan sebuah *instance* dari suatu kelas dengan melakukan pemilihan kelas, pembuatan individu dari kelas, dan pengisian nilai *slot*.

Menurut Abburu & Babu, banyak metodologi rekayasa ontologi memiliki kesamaan langkah dalam mengembangkan ontologi [1] yaitu sebagai berikut:

- 1) Mendefinisikan batasan dan menentukan ruang lingkup dan pertanyaan kompetensi.
- 2) Konseptualisasi: Abstrak kosakata semantik (konsep) dan pernyataan tentang target domain.
- 3) Formalisasi: Menentukan konsep hierarki dan hubungan antara konsep-konsep.

- 4) Implementasi: Membangun ontologi menggunakan IDE seperti Protege dan Neon.
- 5) Evaluasi: Memeriksa konsistensi menggunakan *reasoner* ontologi dan validasi dengan pertanyaan kompetensi yang valid.

Dari beberapa artikel mengenai tahapan pengembangan ontologi, oleh Abburu & Babu seperti pada Gambar 1 disimpulkan bahwa tahapan dalam mengembangkan ontologi terdiri dari 5 tahap yaitu mendefinisikan batasan dan menentukan ruang lingkup dan kompetensi pertanyaan, konseptualisasi, formalisasi, implementasi, dan evaluasi [1]. Menurut [16], metodologi pengembangan ontologi yang mudah bagi pemula yang baru mengenal ontologi adalah *ontology development 101* yang dikenalkan oleh Noy, N.F., & Mc-Guinness, D.L. [10]. Metodologi pengembangan ontologi dari Abburu & Babu [1] dapat direlasikan dengan *ontology development 101* dari Noy, N.F., & Mc-Guinness, D.L. [10].



Gambar 1. Proses pengembangan ontologi [1]

Kombinasi dari kedua metodologi tersebut digunakan dalam memodelkan ontologi pengetahuan terkait layanan TIK sehingga hasil kombinasi adalah sebagai berikut:

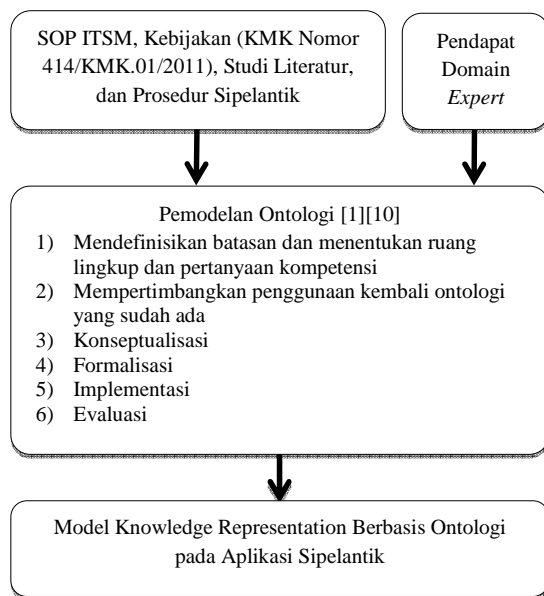
- 1) Mendefinisikan batasan dan menentukan ruang lingkup dan pertanyaan kompetensi.
- 2) Mempertimbangkan penggunaan kembali ontologi yang sudah ada.
- 3) Konseptualisasi: Menentukan istilah penting dalam ontologi.
- 4) Formalisasi: Mendefinisikan kelas dan hierarki kelas, mendefinisikan properti dari kelas-slot, mendefinisikan faset dari slot, membuat *instance*.
- 5) Implementasi: Membangun ontologi menggunakan IDE seperti Protege dan Neon.

- 6) Evaluasi: Memeriksa konsistensi menggunakan *reasoner* ontologi dan validasi dengan pertanyaan kompetensi yang valid.

Tahap kedua dilewati dikarenakan belum pernah dilakukan pembangunan model ontologi terkait layanan pada aplikasi Sipelantik.

Kerangka Pikir

Berdasarkan uraian metodologi di atas maka dapat disusun kerangka pikir dalam melakukan penelitian ini seperti pada Gambar 2.



Gambar 2. Kerangka pikir

Metode Pengumpulan Data

Tahap pengumpulan data dilakukan dengan cara menyebarkan kuesioner kepada 30 respon-

den. Kuesioner disebarakan pada area *service support* dengan memperhatikan keterlibatan dalam pengelolaan aplikasi Sipelantik, rentang usia, jabatan, dan masa kerja serta pernah mengikuti pelatihan atau memiliki sertifikasi ITIL. Setelah melakukan pengumpulan data, tahapan selanjutnya adalah menganalisis data. Pada penelitian ini, untuk menganalisis data menggunakan piranti lunak yaitu *Microsoft Excel*. Analisis data dilakukan dengan menghitung modus. Modus digunakan untuk menyatakan pilihan atau nilai yang mempunyai frekuensi paling banyak terjadi. Dengan kata lain, pilihan terbanyak dari responden dalam mengelompokkan subkelas berdasarkan kelasnya dijadikan dasar untuk menentukan data primer. Hasil dari analisis data akan dikonfirmasi dan divalidasi kepada 3 pakar.

3. Hasil dan Pembahasan

Bagian ini membahas mengenai analisis kebutuhan domain pengetahuan layanan TIK dan perancangan model ontologinya.

Analisis Domain dan Ruang Lingkup Ontologi

Tahap pertama pemodelan ontologi adalah menentukan domain dan ruang lingkup ontologi. Domain ontologi yang akan dikembangkan dalam penelitian ini adalah layanan TIK. Metode yang dilakukan untuk melakukan analisis domain pengetahuan pada penelitian ini adalah metode wawancara dan kuesioner. Metode wawancara digunakan untuk menentukan domain pengetahuan pada aplikasi Sipelantik dan metode kuesioner digunakan untuk menentukan kelas dan hierarki kelas pengetahuan pada aplikasi Sipelantik dan menentukan properti kelas pengetahuan dan batasannya. Sesuai hasil wawancara dengan narasum-

TABEL 1
BEBERAPA ISTILAH PADA DOMAIN LAYANAN TIK DARI APLIKASI SIPELANTIK

No	Istilah	Keterangan
1	Basis Pengetahuan	Basis informasi mengenai gangguan, masalah, <i>known error</i> , dan dokumen lainnya.
2	Permintaan Layanan	Permintaan layanan TIK dan permintaan perubahan.
3	Penanganan Gangguan	Informasi mengenai penanganan gangguan
4	Komponen Konfigurasi	Informasi mengenai komponen yang perlu dikelola dalam menyediakan layanan TIK
5	Katalog Layanan	Informasi mengenai semua layanan TIK
6	Pengelola Layanan	Pegawai yang terlibat dalam manajemen layanan TIK area <i>Service Suport</i> sesuai dengan fungsi atau prosesnya
7	Pengguna Layanan	Pegawai yang meminta dan menggunakan layanan TIK
8	Aplikasi	Informasi mengenai aplikasi/sistem informasi
9	Database	Informasi mengenai basis data
10	Storage	Informasi mengenai <i>storage</i>
11	Infrastruktur	Informasi mengenai infrastruktur layanan TIK
12	Infra_Pendukung	Informasi mengenai pendukung infrastruktur layanan TIK
13	Buku Manual Panduan	Informasi tentang buku manual panduan instalasi dan cara penggunaan aplikasi
14	Dokumen PSI	Informasi tentang dokumen pengembangan sistem informasi seperti <i>user requirement</i> dan dokumen teknis
15	FAQ	Informasi tentang solusi dari permasalahan layanan TIK yang bersifat non teknis

ber serta diperkuat dengan hasil kuesioner oleh 30 responden dan 3 pakar, didapatkan bahwa domain pengetahuan pada aplikasi Sipelantik terdiri dari 6 (enam) kelas pengetahuan yaitu basis pengetahuan, pengelola layanan, permintaan layanan, penanganan gangguan, katalog layanan, dan komponen konfigurasi. Berdasarkan prosedur ITSM, prosedur aplikasi Sipelantik, dokumen *service catalog*, dan KMK Nomor 414/KMK.01/2011 Tentang Kebijakan dan Standar Manajemen Layanan Teknologi dan Komunikasi Area *Service Support* serta menurut pandangan dari 3 pakar, dari keenam kelas pengetahuan tersebut kemudian ditentukan hierarki kelas pengetahuan, properti kelas pengetahuan, dan batasannya.

Tujuan pemodelan ontologi adalah untuk mendapatkan model ontologi yang dapat digunakan untuk merepresentasikan pengetahuan terkait layanan TIK pada aplikasi Sipelantik. Ontologi ini akan digunakan oleh pengguna layanan TIK pada area *service support* dan dikelola oleh Bidang Pengelolaan TIK. Kegunaan ontologi yang dikembangkan antara lain: (a) Pengetahuan terkait permintaan layanan TIK, (b) Pengetahuan terkait gangguan layanan TIK, dan (c) Mengetahui pengetahuan tentang komponen konfigurasi yang terkait dengan layanan TIK. Ontologi dikembangkan dengan menggunakan *tool ontology editor* yaitu Protégé versi 4.3.

Pemodelan Konseptualisasi

Berdasarkan dari kelas pengetahuan yang sudah ditentukan pada tahap sebelumnya, dilakukan pemodelan secara konseptualisasi. Tahap ini menentukan istilah yang mencakup domain layanan TIK. Beberapa istilah yang berhubungan dengan domain pengelolaan layanan TIK dapat dilihat pada TABEL 1.

Implementasi

Berdasarkan domain pengetahuan yang sudah ditentukan, dilakukan pemodelan secara formal untuk mendefinisikan kelas dan hubungan antarkelas. Urutan kegiatan yang dilakukan pada tahap ini adalah mendefinisikan kelas dan hierarki kelas pengetahuan, mendefinisikan relasi atau properti kelas pengetahuan, mendefinisikan batasan-batasan dari relasi atau properti, dan mendefinisikan *instance/individu*. Model formalisasi yang dibangun diformalkan menjadi model ontologi menggunakan perangkat lunak Protege. Adapun hasil dari pemodelan formalisasi yang telah diimplementasikan pada perangkat lunak Protege adalah sebagai berikut:

Kelas Pengetahuan

Kelas dan subkelas yang sudah didefinisikan pada tahap formalisasi, selanjutnya dibentuk suatu

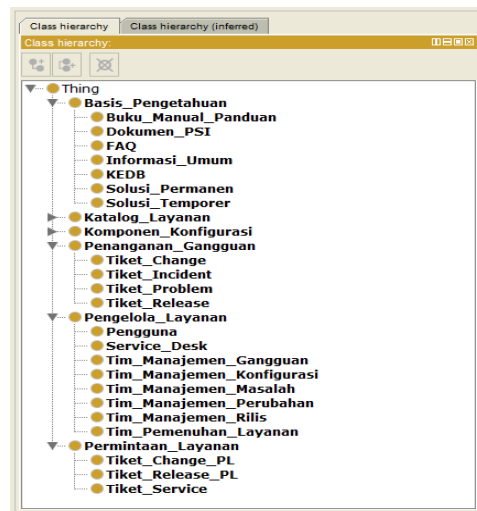
hierarki kelas dan subkelas pada perangkat lunak Protege. Pada Protege untuk mendefinisikan kelas dan hierarki kelas dilakukan pada tab *Classes*. Pada tab *Classes* terdapat *view Class hierarchy*. *View Class hierarchy* digunakan untuk membuat kelas dan subkelas. Hasilnya dapat dilihat pada Gambar 3.

Properti Kelas

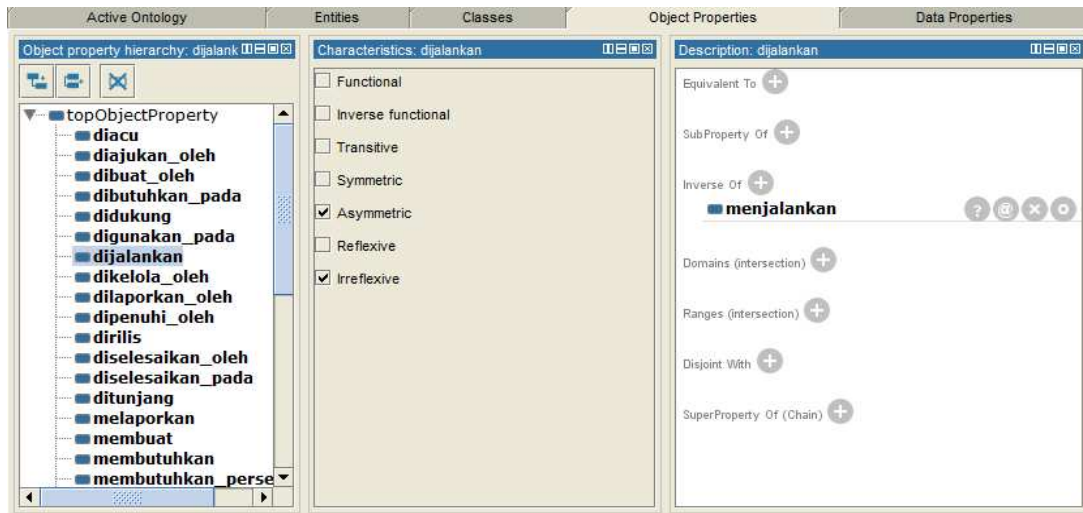
Pada perangkat lunak Protege suatu relasi disebut dengan properti objek dan atribut disebut sebagai properti data. Pada Protege untuk menentukan properti objek dari kelas pengetahuan dilakukan pada tab *Object Properties*. Pada tab *Object Properties* terdapat *view Object Property hierarchy* yang digunakan untuk membuat relasi atau properti objek, *view Characteristics* digunakan untuk menentukan karakteristik dari relasi, dan *view Description* digunakan untuk mendeskripsikan relasi seperti menentukan relasi berlawanan, subjek, dan objek. Properti data dibuat pada tab *Data Properties*. Pada tab *Data Properties* terdapat *view Data Property hierarchy* yang digunakan untuk membuat atribut atau properti data dan *view Description* digunakan untuk mendeskripsikan atribut seperti menentukan tipe data dan kelas.

Hasil implementasi properti objek dan properti data pada Protege adalah model ontologi yang dibangun menghasilkan 37 properti objek dan 100 properti data. Gambar 4 dan Gambar 5 merupakan properti objek dan properti data dari domain layanan TIK pada aplikasi Sipelantik.

Pada Gambar 3 terdapat *thing* yang mewakili domain layanan TIK yang terdiri dari 6 (enam) kelas yaitu kelas basis pengetahuan, pengelola layanan, permintaan layanan, penanganan gangguan, komponen layanan, dan katalog layanan serta diikuti subkelas dari 6 (enam) kelas tersebut



Gambar 3. Kelas dan subkelas pada domain layanan TIK



Gambar 4. Properti objek pada Protege

Individu

Tahap terakhir dari pemodelan formalisasi adalah membuat individu/*instance* dari kelas yang telah didefinisikan. Data yang digunakan sebagai *instance* merupakan data *dummy* yang mewakili dari data pada aplikasi Sipelantik. Pada Protege untuk menentukan individu dari kelas pengetahuan dilakukan pada tab *Individuals*. Pada tab *Individuals* terdapat *view Individuals* yang digunakan untuk membuat *instance*, *view Description* digunakan untuk mendeskripsikan *instance* seperti tipe dari *instance*, dan *view Property assertions* digunakan untuk menentukan komponen atau data dari *instance*. Gambar 6 merupakan contoh pembuatan individu pada perangkat lunak Protege.

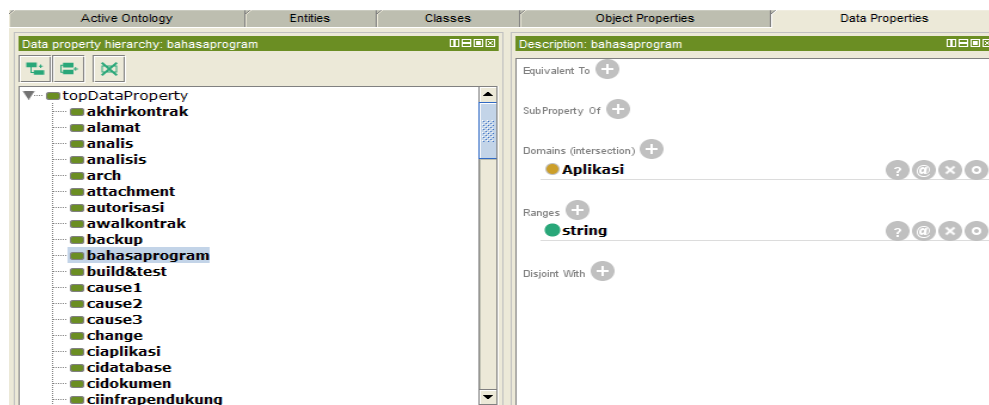
Evaluasi

Pada tahap ini dilakukan uji coba terhadap model ontologi yang telah dibuat. Uji coba dilakukan untuk memvalidasi dan menguji konsistensi logika model ontologi yang telah dibangun. Mo-

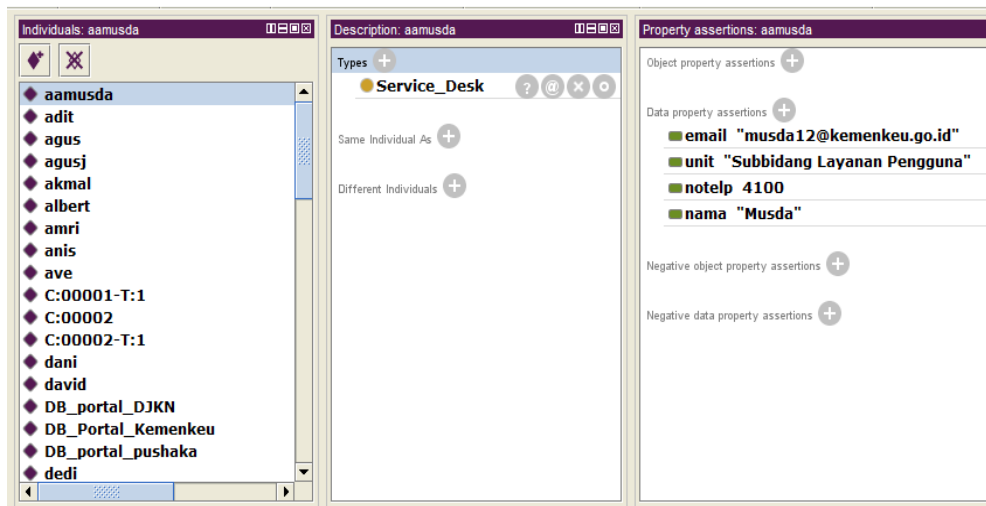
del ontologi direpresentasikan dalam bentuk OWL. Kelas dan subkelas yang berelasi digunakan untuk membentuk pengetahuan, kemudian pengetahuan tersebut direpresentasikan dalam bentuk SPARQL atau DL Query pada perangkat lunak *Protege*. Sebelum menjalankan perintah SPARQL dan DL Query terlebih dahulu dilakukan validasi *reasoning* dengan menggunakan *Pellet*. *Pellet Plugin* ini digunakan untuk menguji ke-konsistensian logika model ontologi yang telah dibangun berdasarkan hierarki kelas dan relasi yang dihasilkan. Berikut ini tahapan uji coba terhadap model ontologi yang telah dibangun:

Uji Coba Model Ontologi dengan Pellet

Pada tahap ini dilakukan validasi terhadap model ontologi yang telah dibangun. Validasi dilakukan untuk mengetahui dan memeriksa ada atau tidaknya kesalahan pada model ontologi. Pada Protege, validasi model ontologi dengan melakukan *reasoning* menggunakan *Pellet*. Vali-



Gambar 5. Properti data pada Protege



Gambar 6. Contoh pembuatan individu pada Protege

dasi ini selain untuk memeriksa konsistensi dari model ontologi juga untuk melakukan proses inferensi. Kelas yang tidak konsisten akan terlihat berwarna merah, sehingga pada model ontologi yang terdapat kelas yang tidak konsisten, kelas tersebut menjadi berwarna merah. Hasil uji coba *reasoning* pada perangkat lunak *Protege* kali ini, tidak menunjukkan adanya kelas yang berwarna merah dan tidak menampilkan tanda peringatan adanya kelas yang tidak konsisten.

Uji Coba dengan Visualization

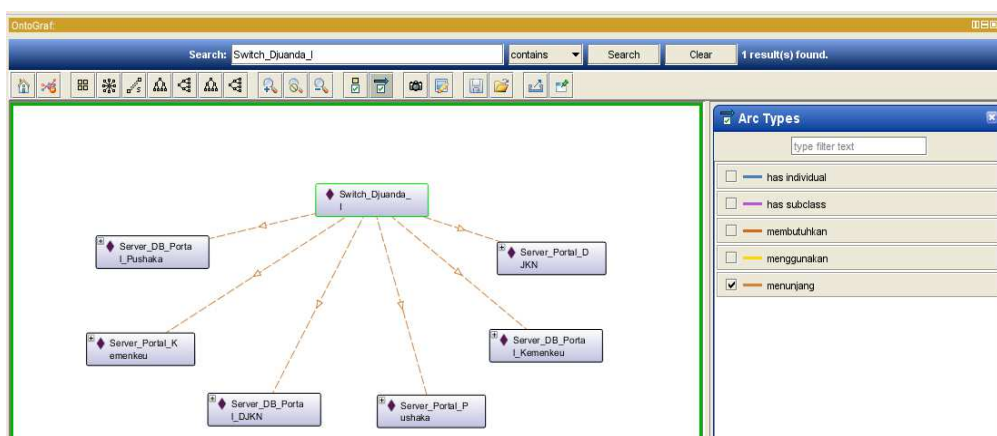
Uji coba visualisasi dilakukan pada perangkat lunak *Protege* dengan menggunakan *plugin OntoGraf*. Uji coba ini dimaksudkan untuk melihat hubungan atau relasi dari suatu *instance*. Pada uji coba kali ini dilakukan pencarian menggunakan satu kata kunci dari sebuah *instance* yaitu "Switch_Djuanda_I", maka pada *OntoGraf* menampilkan *instance* "Switch_Djuanda_I" dan me-

nampilkan kemungkinan relasi yang terbentuk. Pada Gambar 7 menampilkan relasi yang terbentuk dari *instance* "Switch_Djuanda_I".

Pada Gambar 7, relasi atau properti objek yang terbentuk adalah "menunjang". *Instance* "Switch_Djuanda_I" memiliki relasi dengan *instance* lainnya seperti: *Server Portal DJKN*, *Server Portal Pushaka*, *Server Portal Kemenkeu*, *Server DB Portal DJKN*, *Server DB Portal Pushaka*, *Server DB Portal Kemenkeu*. Visualisasi ini membantu para analis atau Pelaksana Teknis sebagai bahan dalam mengambil keputusan terkait permasalahan teknis semisal "Switch_Djuanda_I" terjadi kerusakan.

Uji Coba dengan SPARQL atau DL Query

Uji coba dengan menggunakan *query* dilakukan dengan menggunakan *plugin SPARQL Query* atau *DL Query*. Pengujian dilakukan dengan menguji apakah model ontologi yang dibangun mam-



Gambar 7. Relasi antar individu pada kelas komponen konfigurasi

pu menjawab pertanyaan kompetensi dari tiga pakar. Pengujian dilakukan dengan membuat kueri untuk menjawab 15 pertanyaan kompetensi dari pakar. Terdapat 2 pertanyaan kompetensi yang sama dari dua pakar, sehingga jumlah pertanyaan kompetensi yang diujicobakan menjadi 13 pertanyaan. Adapun hasil pengujian pada 13 pertanyaan dapat dilihat pada TABEL 2.

TABEL 2
TABULASI HASIL EVALUASI MODEL ONTOLOGI

No	Pertanyaan	Hasil
1	Apa <i>known error</i> untuk Portal DJKN gagal ubah data?	1
2	Berapa lama waktu yang diperlukan untuk menyelesaikan tiket incident I:00001?	1
3	Komponen konfigurasi apa saja yang ditunjang Switch_Djuanda_I?	1
4	Dimana lokasi database portal pushaka?	1
5	Apa solusi temporer untuk gangguan pada Portal DJKN tidak dapat edit data?	1
6	Solusi apa untuk menyelesaikan gangguan pada Portal DJKN tidak bisa ubah data pada tiket problem?	1
7	Aplikasi Portal DJKN berhubungan dengan komponen konfigurasi apa saja?	1
8	Aplikasi Portal DJKN termasuk dalam kategori layanan apa pada service catalog?	1
9	Siapa yang mengajukan permintaan hosting portal DJKN?	1
10	Bagaimana mengatasi gangguan komputer tidak dapat masuk os?	1
11	Apa saja komponen konfigurasi yang berhubungan dengan gangguan pada portal DJKN tidak bisa ubah data?	1
12	Siapa analis yang melakukan penyelesaian permintaan hosting aplikasi Portal DJKN?	1
13	Apa solusi permanen untuk gangguan pada Portal DJKN tidak bisa ubah data?	1

Keterangan:

1 = sesuai 0 = tidak sesuai

Dari 13 pertanyaan kompetensi, setelah dilakukan uji coba dengan menggunakan SPARQL Query atau DL Query model ontologi yang dibangun dapat menjawab semua pertanyaan tersebut dan model ontologi tepat untuk merepresentasikan pengetahuan terkait layanan TIK pada aplikasi Sipelantik.

4. Kesimpulan

Model ontologi yang tepat untuk merepresentasikan pengetahuan dalam domain layanan TIK pada aplikasi Sipelantik di Pusintek Kementerian Keuangan menghasilkan 6 (enam) kelas pengetahuan yaitu Basis Pengetahuan, Pengelola Layanan, Permintaan Layanan, Penanganan Gangguan, Katalog Layanan, dan Komponen Konfigurasi. Kelas Basis Pengetahuan terdiri dari 7 subkelas, kelas Pengelola Layanan terdiri dari 8 subkelas, kelas Permintaan Layanan terdiri dari 3 subkelas, kelas Penanganan Gangguan terdiri dari 4 subkelas, kelas Komponen Konfigurasi terdiri

dari 7 subkelas, dan kelas Katalog Layanan terdiri dari 20 subkelas. Model ontologi yang dibangun juga menghasilkan 37 properti objek dan 100 properti data. Uji konsistensi terhadap model ontologi dengan *Pellet Reasoner* menunjukkan bahwa model ontologi yang dibangun konsisten karena hubungan antar konsep didefinisikan dengan benar. Pengujian dengan uji query SPARQL atau DL Query telah berhasil melakukan pencarian dan pengolahan data terhadap hasil ontologi dengan berbagai variasi query berdasarkan 13 buah pertanyaan kompetensi.

5. Saran

Melalui penelitian ini, saran untuk penelitian selanjutnya dapat diperkuat lagi dengan memperjelas kesesuaian properti kelas pengetahuan, menambahkan domain pengetahuan layanan TIK lainnya yang belum termasuk dalam penelitian ini, dan mengembangkan model ontologi ke dalam bentuk *semantic web* dengan membuat aplikasi berbasis *web* sehingga memudahkan untuk melakukan modifikasi atau penambahan *class*, *slot*, dan *instance* serta pencarian *query*.

Referensi

- [1] Abburu, S. & Babu, G. S., "A Framework for Ontology Based Knowledge Management", *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, pp. 2231-2307. 2013.
- [2] Antoniou G, & Hermalen F., *A Semantic Web Primer*, The MIT Press, Cambridge (GB), P.259, 2004.
- [3] Daconta, M.C., Obsrt, L.J., & Smith, K.T., *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, Wiley Publishing Inc, Indianapolis, p.304 2003.
- [4] Gomez-Perez, A., & Benjamins, V.R., *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*, 1999
- [5] Haisheng Li, Wenzheng Li, Qiang Cai, & Hongzhi Liu, "A Framework of Ontology-based Knowledge Management System", *In Proc. IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, pp. 374-377, 2009.
- [6] Hua Wang, "Research on the Model of Knowledge Representation Ontology Based on Framework in Intelligent Learning System", *IEEE International Conference on Electrical and Control Engineering (ICECE)* pp. 6757-6760, 2011.

- [7] Ji Ma, Xinping Pan, & Jiayi Yao, "The Research on the Knowledge Representation Model of XBRL", *IEEE International Conference on Service Operations and logistics, and Informatics*, Vol. 1, pp. 734-739, 2008.
- [8] Kishore, R., Sharman, R., & Ramesh, R., "Computational Ontologies and Information Systems: I. Foundations", *Communications of the Association for Information Systems (CAIS)*, Vol. 14, pp.158-183, 2004.
- [9] Kougiyas, I., Seremeti, L., & Kalogeras, D., "Ontology-Based Knowledge in NGEES. *International Journal of Pure and Applied Science and Management Technology*, Vol. 2, pp. 54-62, 2011.
- [10] Noy, N.F., & McGuinness, D.L., "Ontology Development 101: A Guide to Creating Your First Ontology", *Knowledge System Laboratory (KSL) of Department of Computer Science Stanford USA: Technical Report, KSL-01-05*, 2001.
- [11] Smith, B., "Ontology and Information Systems. *Blackwell Guide to the Philosophy of Computing and Information*, pp. 155-166, 2003.
- [12] Subhashini, R., & Akilandeswari, J., "A Survey on Ontology Construction Methodologies", *International Journal of Enterprise Computing and Business Systems*, Vol.1, pp. 1-14, 2011.
- [13] Tanwar, P., Prasad, T. V., & Aswal, M. S., "Comparative Study of Three Declarative Knowledge Representation Techniques", *International Journal on Computer Science and Engineering (IJCSSE)*, Vol. 2, pp.2274-2281, 2010.
- [14] Yan Lei, Wang Xinying, & Dong Junlei, "A Power Grid Knowledge Representation Using Agent Based Knowledge Representation in Pervasive Computing", *The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, Vol. 2, pp. 297-300, 2010.
- [15] Horridge, M. *A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.3*. The University Of Manchester, 2011.
- [16] Cristani, M. and R. Cuel, "A Survey on Ontology Creation Methodologies. *Int. J. Semantic Web Inform. Syst.*, Vol. 1(2), pp. 46-69, 2005.

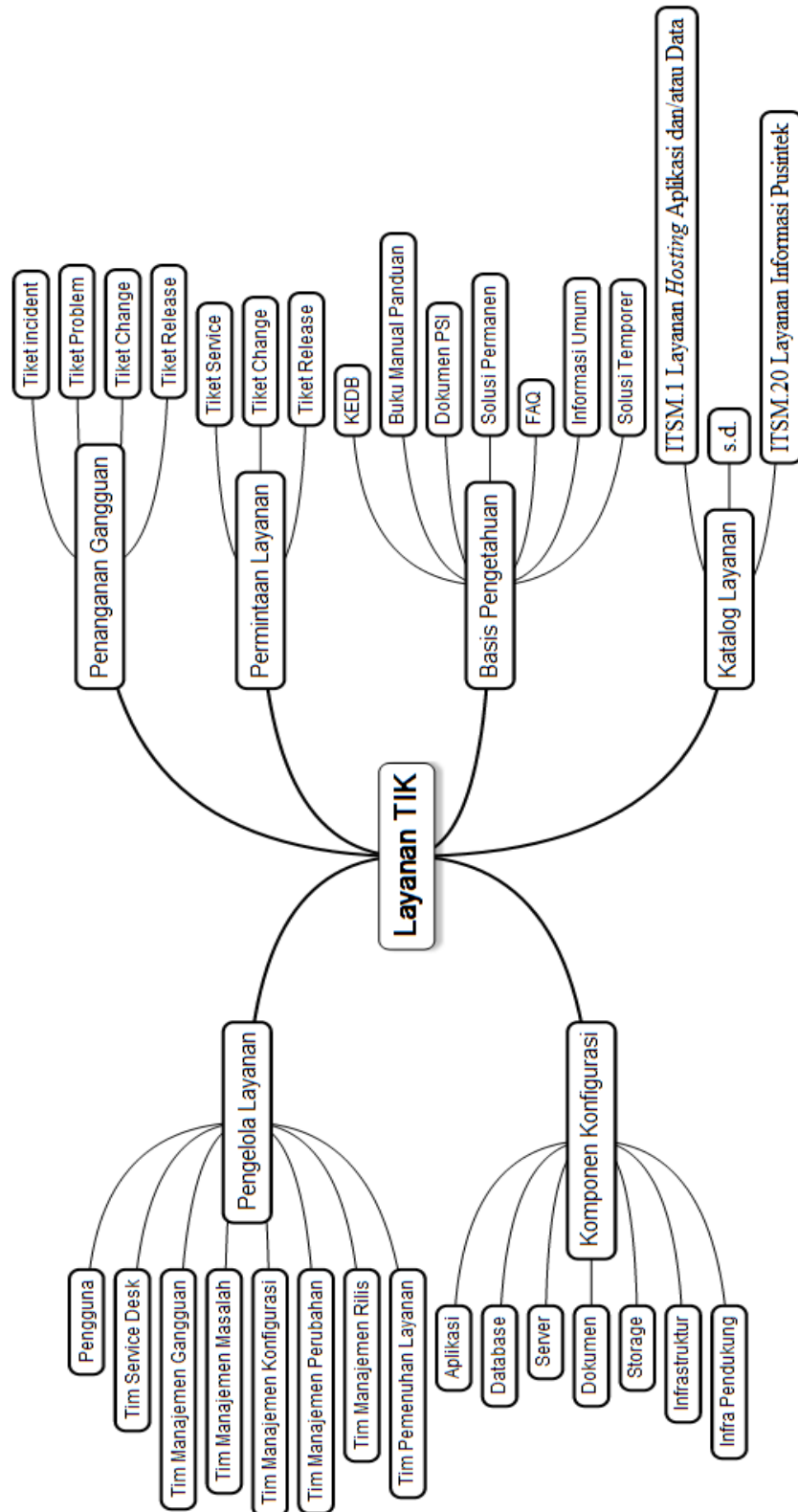
Lampiran

Lampiran 1

TABEL
KARAKTERISTIK RELASI ANTARKELAS

No	Subjek	Predikat	Objek	Relasi Invers	Karakteristik
1	Pengelola Layanan	Membuat	Basis Pengetahuan	Dibuat oleh	<i>Asymmetric, Irreflexive</i>
2	Pengelola Layanan	Memenuhi	Permintaan Layanan	Dipenuhi oleh	<i>Asymmetric, Irreflexive</i>
3	Pengelola Layanan	Menyelesaikan	Penanganan Gangguan	Diselesaikan oleh	<i>Asymmetric, Irreflexive</i>
4	Pengelola Layanan	Mengelola	Komponen Konfigurasi	Dikelola oleh	<i>Asymmetric, Irreflexive</i>
5	Permintaan Layanan	Mengacu	Katalog Layanan	Diacu	<i>Asymmetric, Irreflexive</i>
6	Basis Pengetahuan	Membutuhkan	Komponen Konfigurasi	Dibutuhkan pada	<i>Asymmetric, Irreflexive</i>
7	Permintaan Layanan	Membutuhkan	Komponen Konfigurasi	Dibutuhkan pada	<i>Asymmetric, Irreflexive</i>
8	Penanganan Gangguan	Membutuhkan	Komponen Konfigurasi	Dibutuhkan pada	<i>Asymmetric, Irreflexive</i>
9	Katalog Layanan	Menggunakan	Komponen Konfigurasi	Digunakan pada	<i>Asymmetric, Irreflexive</i>
10	Pengelola Layanan	Memenuhi	Katalog Layanan	Dipenuhi oleh	<i>Asymmetric, Irreflexive</i>

Lampiran 2



Gambar Hierarki kelas pada domain layanan TIK