

Inconsistency Measure OWL Ontology Berbasis Axiom dengan MinInc Inconsistency Value

Nur Alfi Ekowati¹, Sunaryono², dan Dwi Prasetyo³

^{1,2,3}STMIK Widya Utama

¹nuralfiekowati@gmail.com, ²aryo.jateng@gmail.com, ³dwiprasetyo919@gmail.com

Abstract— Inconsistency measure adalah sebuah pengukur yang digunakan untuk menganalisis inkonsistensi pada knowledge base. Pengukur ini sangat berguna untuk menganalisis tingkat keparahan inkonsistensi dari sebuah ontology. Pengukur inkonsistensi (inconsistency measure) tersebut dapat diterapkan di era kini guna membantu proses analisis data dalam jumlah besar.

Saat ini ada beberapa inconsistency measure untuk Propositional Logic yang telah tersedia. Sebuah gagasan menarik muncul dan perlu diteliti, yaitu kemungkinan diciptakannya inconsistency measure berbasis axiom dengan MinInc Inconsistency Value (MIV) untuk OWL ontology serta mengadaptasi atau mengkonversikan measure-measure Propositional Logic tersebut. Tiga measure untuk OWL ontology telah diciptakan secara teoritis, yaitu MIV_D , $MIV_{\#}$, dan MIV_C . Selain itu pembuatan program aplikasi ketiga measure tersebut juga dilakukan sebagai implementasinya dengan menggunakan OWL reasoner tertentu. Sehingga measure tersebut dapat lebih dirasakan manfaatnya secara praktis. Nilai inkonsistensi akan memberi intuisi untuk menggali cara dalam menangani inkonsistensi ontology tersebut.

Metode pengumpulan data pada penelitian ini adalah melalui studi pustaka. Sedangkan metode pengembangan sistem dari penelitian ini menggunakan metode waterfall dengan tahapan-tahapan yang ada pada SDLC (Software Development Life Cycle) untuk membangun sebuah perangkat lunak. Dari ketiga measure yang ada, MIV_D adalah inconsistency measure paling sederhana perhitungannya dan menghasilkan nilai inkonsistensi antara 0 dan 1 saja.

Keywords—3-5 inconsistency measure, ontology, axiom, MinInc Inconsistency Value

1. PENDAHULUAN

Menurut Krötzsch dkk (2012), “Description Logics are a family of knowledge representation languages that are widely used in ontological

modeling”. Description Logic (DL) biasa digunakan untuk membuat aplikasi kecerdasan buatan dan sejumlah aplikasi tertentu, seperti aplikasi pada bidang biomedical informatics. Knowledge base (kumpulan informasi, disingkat KB) yang terkumpul menjadi satu kesatuan secara terminologi untuk merepresentasikan sebuah kasus pada era kini biasa disebut ontology. Bahasa ontology OWL 2 adalah bahasa yang berdasarkan DL SROIQ, dimana bahasa tersebut adalah bahasa DL yang paling ekspresif dan umum digunakan saat ini. Rudolph (2011) menyebutkan bahwa “most of mainstream DLs today are, in fact, sublanguages of DL SROIQ”. Secara umum ontology memiliki peranan penting dalam pembangunan web semantik yang merupakan ekstensi dari website.

Ontology pada aplikasi seringkali dikompilasi oleh lebih dari satu orang. Hal ini memungkinkan adanya data yang tidak konsisten. Ontology yang tidak konsisten dapat merusak informasi yang diakses untuk tujuan tertentu. Selain itu, tingkat keparahan yang tinggi dari inkonsistensi dapat mengakibatkan masalah serius pada kasus tertentu. Jika hal itu terjadi maka ontology yang tidak konsisten tersebut harus diperbaiki.

Secara umum, peran inconsistency measure juga menjadi begitu penting untuk memastikan bahwa semua informasi dalam skala besar pada sebuah ontology telah konsisten. Hal tersebut dikarenakan pengecekan secara manual akan memakan lebih banyak waktu dan tenaga, serta tingkat keakuratannya pun menjadi dipertanyakan. Selain itu, tingkat keparahan inkonsistensi suatu ontology yang dihasilkan oleh inconsistency measure akan memberi pengaruh pada bagaimana cara penanganannya. Nilai inkonsistensi akan memberi intuisi untuk menggali cara dalam menangani inkonsistensi ontology. Perbedaan level nilai antar ontology memungkinkan adanya perbedaan cara penanganannya.

Tujuan dari penelitian ini adalah sebagai berikut menciptakan inconsistency measure untuk OWL ontology yang dapat diterapkan di era kini guna membantu proses analisis data dalam jumlah besar. Dalam hal ini measure dibangun dengan mengadaptasi atau mengkonversi inconsistency measure untuk Propositional Logic yang telah ada.

2. METODE PENELITIAN

2.1. Metode Pengumpulan Data

Metode pengumpulan data pada penelitian ini dilakukan melalui teknik penelitian utama, yaitu studi pustaka. Dalam penelitian ini, pengumpulan data utamanya dilakukan dengan cara mempelajari beberapa *paper* yang terkait dengan topik penelitian. Beberapa diantaranya tercantum dalam daftar pustaka.

2.2. Metode Pengembangan Sistem

Metode pengembangan sistem dari penelitian ini menggunakan metode *waterfall*. Metode tersebut dilakukan dengan mengikuti tahapan-tahapan yang ada pada SDLC (*Software Development Life Cycle*), meliputi analisis kebutuhan (*requirements*), analisis sistem (*analysis*), perancangan (*design*), implementasi (*coding/implementation*), pengujian (*testing*), dan perawatan (*maintenance*).

3. HASIL DAN PEMBAHASAN

3.1. Inconsistency Measures OWL Ontology

Pada pokok pembahasan ini *inconsistency measure* untuk OWL 2 akan didefinisikan dan diuraikan. *Inconsistency measure* tersebut adalah *measure* pilihan yang diadaptasi (ditransfer) dari *Propositional Logic*. Pembahasan terbatas untuk *inconsistency measure* berbasis *axiom* saja. Untuk menggambarkan *measure-measure* tersebut, *knowledge base* \mathcal{K}_1 , \mathcal{K}_2 , dan \mathcal{K}_3 berikut digunakan sebagai contoh.

Contoh 1. Katakanlah $N_C := \{ProjectManager, Employee, Developer, Person\}$, $N_I := \{alfi, dewi\}$, $N_R := \{manage\}$ dan *knowledge base* \mathcal{K}_1 , \mathcal{K}_2 , dan \mathcal{K}_3 didefinisikan sebagai berikut:

$$\begin{aligned} \mathcal{K}_1 &= \{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, \\ &\quad ProjectManager(alfi)\}, \\ \mathcal{K}_2 &= \{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, \\ &\quad ProjectManager(alfi), ProjectManager(dewi)\}, \\ \mathcal{K}_3 &= \{ProjectManager \sqsubseteq Employee \sqcap \neg manage, Developer, Employee \sqsubseteq Developer \sqcap Person, \\ &\quad ProjectManager(alfi)\}. \end{aligned}$$

\mathcal{K}_1 dan \mathcal{K}_2 adalah contoh representasi *knowledge base* inkonsisten dengan jumlah *class assertion* yang berbeda dan berdampak pada tingkat inkonsistensi yang berbeda pula. Sementara itu \mathcal{K}_3 adalah representasi *knowledge base* yang konsisten. Secara umum, semakin besar nilai inkonsistensi yang dihasilkan dari sebuah *measure* berarti semakin besar pula inkonsistensi dari *knowledge base* tersebut. Di sisi lain, *knowledge base* yang konsisten akan mempunyai nilai 0 sebagai nilai inkonsistensinya. Untuk

mengadaptasi *notion* dari *inconsistency measure* terpilih, formula pada *Propositional Logic* diterjemahkan sebagai sebuah *axiom* pada *ontology*, sementara *atom* diterjemahkan sebagai *signature* (termasuk di dalamnya *top* dan *bottom concept*), dilambangkan dengan $Sign \cup \{\top, \perp\}$.

1.2. Inconsistency Measure Berbasis Axiom dengan MinInc Inconsistency Value (MIV)

Penelitian ini mendapatkan tiga *inconsistency measure* untuk OWL 2 yang terpilih dan masuk dalam kategori *measure* berbasis *axiom*.

Definisi 1 (MinInc Inconsistency Value (MIV)). Sebuah MinInc Inconsistency Value (MIV) adalah sebuah fungsi $MIV(\mathcal{K}, \alpha)$ yang memetakan pasangan (\mathcal{K}, α) ke elemen $\mathbb{R}_{\geq 0}^{\infty}$, dimana *knowledge base* $\mathcal{K} \in \mathbb{K}$ dan $\alpha \in \mathcal{K}$.

Hunter and S. Konieczny (2008) mengajukan satu grup *inconsistency measure* berdasarkan *MinInc Inconsistent Value* yang didesain untuk *Propositional Logic*. *Measure* tersebut mendefinisikan *inconsistency value* yang memberikan inkonsistensi setiap *axiom* pada *knowledge base*. Pada pembahasan berikut, seluruh *inconsistency measure* akan disebut MIV *inconsistency measure* dimana MIV adalah kependekan dari *MinInc Inconsistency Value*.

1.2.1. MIV_D Inconsistency Measure

Definisi 2 (free axiom). $MI(\mathcal{K})$ adalah sebuah set dari minimal inconsistent subset. Sebuah free axiom dari *knowledge base* \mathcal{K} adalah *axiom* dari \mathcal{K} yang bukan milik subset $MI(\mathcal{K})$ manapun.

MIV_D *inconsistency measure* adalah sebuah *drastic measure* yang memberi nilai 1 jika sebuah *axiom* adalah milik *minimal inconsistent subsets*. Jika tidak maka nilainya adalah 0. Hal itulah yang membedakan antara *free axiom* dengan lainnya.

Definisi 3 (MIV_D inconsistency measure). Katakanlah $\mathcal{K} \in \mathbb{K}$ dan $\alpha \in \mathcal{K}$. MIV_D inconsistency measure: $(\mathcal{K}, \alpha) \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ didefinisikan sebagai berikut:

$$MIV_D(\mathcal{K}, \alpha) = \begin{cases} 1, & \text{if } \exists M \in MI(\mathcal{K}), \alpha \in M; \\ 0, & \text{otherwise.} \end{cases}$$

Contoh 2. Dengan meneruskan Contoh 1, akan didapatkan $MI(\mathcal{K}_1)$, $MI(\mathcal{K}_2)$, and $MI(\mathcal{K}_3)$ sebagai berikut:

$$MI(\mathcal{K}_1) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\},$$

$$MI(\mathcal{K}_2) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}, \{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\},$$

$$MI(\mathcal{K}_3) = \emptyset.$$

Maka

$$MIV_D(\mathcal{K}_1, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) = 1,$$

$$MIV_D(\mathcal{K}_1, Employee \sqsubseteq Developer \sqcap Person) = 1,$$

$$MIV_D(\mathcal{K}_1, ProjectManager(alfi)) = 1.$$

$$MIV_D(\mathcal{K}_2, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) = 1,$$

$$MIV_D(\mathcal{K}_2, Employee \sqsubseteq Developer \sqcap Person) = 1,$$

$$MIV_D(\mathcal{K}_2, ProjectManager(alfi)) = 1,$$

$$MIV_D(\mathcal{K}_2, ProjectManager(dewi)) = 1.$$

Karena $MI(\mathcal{K}_3)$ adalah *empty set* (set kosong), maka

$$MIV_D(\mathcal{K}_3, ProjectManager \sqsubseteq Employee \sqcap \neg \exists manage.Developer) = 0,$$

$$MIV_D(\mathcal{K}_3, Employee \sqsubseteq Developer \sqcap Person) = 0,$$

$$MIV_D(\mathcal{K}_3, ProjectManager(alfi)) = 0.$$

1.2.2. $MIV_{\#}$ Inconsistency Measure

Measure kedua berikut merupakan salah satu jenis *MIV inconsistency measure* yang sederhana karena hanya mengambil nilai kardinalitas yang menghitung jumlah *minimal inconsistent subsets* dimana *axiom* yang bersangkutan adalah milik subset tersebut.

Definisi 4 ($MIV_{\#}$ inconsistency measure). *Katakanlah* $\mathcal{K} \in \mathbb{K}$, $\alpha \in \mathcal{K}$. $MIV_{\#}$ inconsistency measure: $(\mathcal{K}, \alpha) \rightarrow \mathbb{R}_{\geq 0}^{\infty}$

didefinisikan sebagai berikut:

$$MIV_{\#}(\mathcal{K}, \alpha) = |\{M \in MI(\mathcal{K}) \mid \alpha \in M\}|.$$

Penggunaan $MIV_{\#}$ *inconsistency measure* akan ditunjukkan pada contoh berikut.

Contoh 3. Dengan meneruskan Contoh 1 dan memanggil kembali $MI(\mathcal{K}_1)$, $MI(\mathcal{K}_2)$, and $MI(\mathcal{K}_3)$ seperti pada Contoh 2:

$$MI(\mathcal{K}_1) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\},$$

$$MI(\mathcal{K}_2) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}, \{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\},$$

$$MI(\mathcal{K}_3) = \emptyset.$$

Maka akan didapatkan:

$$MIV_{\#}(\mathcal{K}_1, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) = 1,$$

$$MIV_{\#}(\mathcal{K}_1, Employee \sqsubseteq Developer \sqcap Person) = 1,$$

$$MIV_{\#}(\mathcal{K}_1, ProjectManager(alfi)) = 1.$$

$$MIV_{\#}(\mathcal{K}_2, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) = 2,$$

$$MIV_{\#}(\mathcal{K}_2, Employee \sqsubseteq Developer \sqcap Person) = 2,$$

$$MIV_{\#}(\mathcal{K}_2, ProjectManager(alfi)) = 1,$$

$$MIV_{\#}(\mathcal{K}_2, ProjectManager(dewi)) = 1.$$

$$MIV_{\#}(\mathcal{K}_3, ProjectManager \sqsubseteq Employee \sqcap \neg \exists manage.Developer) = 0,$$

$$MIV_{\#}(\mathcal{K}_3, Employee \sqsubseteq Developer \sqcap Person) = 0,$$

$$MIV_{\#}(\mathcal{K}_3, ProjectManager(alfi)) = 0.$$

1.2.3. MIV_C Inconsistency Measure

Inconsistency measure ketiga pada kelompok *MIV measure* berikut akan memperhitungkan kardinalitas anggota *minimal inconsistent subset* dimana *axiom* yang bersangkutan adalah anggota yang dimaksud. Ide dari *measure* tersebut kemudian didefinisikan pada definisi berikut.

Definisi 5 (MIV_C inconsistency measure). *Katakanlah* $\mathcal{K} \in \mathbb{K}$, $\alpha \in \mathcal{K}$. MIV_C inconsistency measure: $(\mathcal{K}, \alpha) \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ *didefinisikan sebagai berikut:*

$$MIV_C(\mathcal{K}, \alpha) = \sum_{M \in MI(\mathcal{K}) \text{ s.t. } \alpha \in M} \frac{1}{|M|}.$$

Berdasarkan definisi tersebut, MIV_C *inconsistency measure* akan menjumlahkan seluruh perhitungan pembagian dari kardinalitas subset-subset yang ada. Perhitungan tersebut akan diterapkan pada contoh berikut.

Contoh 4. Dengan meneruskan Contoh 1 dan memanggil kembali $MI(\mathcal{K}_1)$, $MI(\mathcal{K}_2)$, and $MI(\mathcal{K}_3)$ seperti pada Contoh 2:

$$MI(\mathcal{K}_1) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\},$$

$$MI(\mathcal{K}_2) = \{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}, \{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\},$$

$$MI(\mathcal{K}_3) = \emptyset.$$

Maka akan didapatkan:

$$MIV_C(\mathcal{K}_1, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) =$$

$$\frac{1}{|\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}|} = \frac{1}{3}.$$

$$MIV_C(\mathcal{K}_1, Employee \sqsubseteq Developer \sqcap Person) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}\}}|} = \frac{1}{3},$$

$$MIV_C(\mathcal{K}_1, ProjectManager(alfi)) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}\}}|} = \frac{1}{3},$$

$$MIV_C(\mathcal{K}_2, ProjectManager \sqsubseteq Employee \sqcap \neg Developer) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}\}}|} +$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\}\}}|} =$$

$$= \frac{1}{3} + \frac{1}{3} = \frac{2}{3},$$

$$MIV_C(\mathcal{K}_2, Employee \sqsubseteq Developer \sqcap Person) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}\}}|} +$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\}\}}|} =$$

$$= \frac{1}{3} + \frac{1}{3} = \frac{2}{3},$$

$$MIV_C(\mathcal{K}_2, ProjectManager(alfi)) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(alfi)\}\}}|} = \frac{1}{3},$$

$$MIV_C(\mathcal{K}_2, ProjectManager(dewi)) =$$

$$\frac{1}{|\{\{ProjectManager \sqsubseteq Employee \sqcap \neg Developer, Employee \sqsubseteq Developer \sqcap Person, ProjectManager(dewi)\}\}}|} = \frac{1}{3},$$

$$MIV_C(\mathcal{K}_3, ProjectManager \sqsubseteq Employee \sqcap \neg \exists manage.Developer) = 0,$$

$$MIV_C(\mathcal{K}_3, Employee \sqsubseteq Developer \sqcap Person) = 0,$$

$$MIV_C(\mathcal{K}_3, ProjectManager(alfi)) = 0.$$

4. KESIMPULAN

Penelitian ini berhasil membuat 3 *inconsistency measure* yang diadaptasi dari *Propositional Logic*. Keseluruhan *measure* tersebut adalah *inconsistency measure* berbasis *axiom* dan disebut *MIV inconsistency measure*. Ketiga *measure* tersebut adalah *MIV_D inconsistency measure*, *MIV_# inconsistency measure*, dan *MIV_C inconsistency measure*.

MIV_D inconsistency measure adalah *measure* paling sederhana dan merupakan sebuah *drastic measure* yang memberi nilai 1 jika sebuah *axiom* adalah inkonsisten, jika tidak maka nilainya 0. Sedangkan *MIV_#* dan *MIV_C inconsistency measure* adalah *measure* yang memanfaatkan kardinalitas dari elemen yang terkait dengan *axiom* pada *ontology*.

DAFTAR PUSTAKA

- [1] Krötzsch, Markus, František Simančík, dan Ian Horrocks. 2012. *A Description Logic Primer*. Technical Report, *arXiv.org*, volume CoRR abs/1201.4089.
- [2] Rudolph, Sebastian. 2011. *Foundations of Description Logics*. Reasoning Web 2011, pp. 76-136, 2011.
- [3] Hunter, Anthony dan Sebastien Konieczny. 2008. *Measuring Inconsistency Through Minimal*

Inconsistency Sets. Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008), AAAI Press, pp. 358-366.

- [4] Hunter, Anthony dan Sebastien Konieczny. 2006. *Shapley Inconsistency Values*. Proceedings of the 10th International Conference on Knowledge Representation (KR '06), AAAI Press, pp. 249-259.
- [5] Hunter, Anthony dan Sebastien Konieczny. 2004. *Approaches to Measuring Inconsistent Information*. Inconsistency Tolerance, in: Lect. Notes Comput. Sci., vol. 3300, Springer, pp. 189-234.