

DETEKSI KONFLIK LEKSIKAL PADA DIAGRAM KELAS MENGUNAKAN MODIFIKASI GRAF DAN SIMILARITAS WORDNET

Billy Montolalu¹⁾, Siti Rochimah²⁾

^{1,2)} Teknik Informatika, Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
e-mail: billy.montolalu@gmail.com¹⁾, siti@its-sby.edu²⁾

Abstrak

Pada lingkungan pengembangan perangkat lunak banyak pengguna, konflik dapat terjadi karena pengguna melakukan perubahan pada bagian yang sama. Lingkungan pengembangan banyak pengguna menggunakan kontrol versi untuk mengelola perubahan yang terjadi. Ketika salah satu pengguna melakukan penyimpanan, kontrol versi mendeteksi terjadinya konflik. Pendeteksian konflik dilakukan untuk mencegah kloning dan kesalahan sintaksis pada proses penggabungan dua buah versi. Pada penelitian ini digunakan sebuah metode untuk mendeteksi konflik secara leksikal pada diagram kelas dalam lingkungan pengembangan banyak pengguna. Metode ini menggunakan pendekatan modifikasi graf dan similaritas WordNet. Pendekatan modifikasi graf digunakan untuk mencari bagian dalam diagram kelas yang hanya mengalami perubahan. Kelas, atribut, operasi dan relasi dimodelkan menjadi bentuk graf. Pendekatan ini menghasilkan graf dengan aturan minimal. Graf dengan aturan minimal adalah graf yang hanya mengalami perubahan. Konflik dapat dideteksi dengan membandingkan similaritas WordNet dua buah versi graf dengan aturan minimal. Pasangan bagian yang mempunyai nilai similaritas tinggi teridentifikasi sebagai kasus konflik leksikal. Dalam penelitian ini metode diujikan dengan menggunakan diagram kelas yang terdiri dari 30 versi. Dari penelitian ini pendekatan yang diambil diperoleh nilai kappa sebesar 0,6404 yang menunjukkan bahwa hasil tingkat kecocokan antara sistem dan pakar pada penelitian ini adalah kuat (good).

Kata Kunci: kontrol versi model, deteksi konflik, konflik leksikal

Abstract

In multi-user software development environment, conflict occurs when more than one user make a change in the same part of the program. Multi-user software development environment uses version control to manage every changes. When one user make a change, version control check for conflict. The Conflict Checking is usefull for preventing cloning and sintax error in multi-version development process. In this research, a method is proposed for detecting conflict by checking class diagram from multi-user development environment lexically. The proposed method uses modified graf and WordNet Similarity approach. The modified graf is used for searching the changed part in class diagram. Class, attribute, operation, and relation are modeled as a graf. This approach produces a graf with minimum rule. A minimum rule graf is a graf that contains only the changes. Conflict can be detected by comparing WordNet similarization and 2 version of the graf with minimum rule. The one with higer similarity value is detected as lexical conflict case. The proposed method is tested using class diagram which has more than 30 version. The result shows that the method obtains the kappa value 0.6404. This value indicates that the compatibiltiy between the method and expert is strong.

Keywords: model version control, conflict detection, lexicon conflict.

1. PENDAHULUAN

Pada industri rekayasa kebutuhan artefak perangkat lunak seperti kode sumber, dokumentasi, dan artefak lainnya disimpan kedalam suatu repositori. Repositori ini bisa diakses oleh banyak pengguna dan menyimpan semua perubahan yang ada pada artefak. Artefak kode sumber biasanya disimpan dalam suatu

repositori khusus yang dinamakan dengan VCS (*Version Control System*). Dengan adanya VCS ini semua perubahan yang ada didalam kode sumber akan disimpan kedalam suatu versi-versi dan ada manajemen konflik jika ada pengguna melakukan perubahan di kode sumber yang sama. Dengan adanya VCS ini pengguna bisa berkolaborasi dan bekerja secara tim untuk membuat suatu perangkat lunak.

Sama halnya seperti kode sumber, artefak model seperti diagram pengguna, diagram kelas dan diagram aktifitas dibangun oleh banyak pengguna dan sering mengalami perubahan. Oleh karena itu manajemen konflik untuk artefak model juga diperlukan dalam pengembangan perangkat lunak.

VCS dalam kode sumber menggunakan perbandingan berbasis teks untuk manajemen versi dan konflik. Meskipun artefak model disimpan kedalam teks, pendekatan yang digunakan pada manajemen konflik VCS untuk kode sumber tidak akan sesuai jika diterapkan untuk artefak model. Hal ini dikarenakan seorang pemrogram akan lebih familiar dengan sintaksis berbasis teks dibandingkan dengan seorang pemodel yang terbiasa dengan sintaksis berbasis grafis (Brosch dkk, 2011). Disamping itu jika VCS pada kode sumber diterapkan untuk model, struktur graf pada model akan rusak dan informasi sintaksis dan semantik akan hilang (Taentzer dkk, 2010). Oleh karena itu perlu pendekatan khusus untuk menangani manajemen konflik pada model.

Beberapa penelitian telah dilakukan terkait dengan deteksi konflik pada model. Penelitian yang dilakukan oleh Brosch (Broch dkk, 2011) adalah membuat bahasa pemodelan untuk visualisasi konflik dengan menggunakan UML Profile. Pada penelitian tersebut konflik dimodelkan secara visual dan visualisasi ini digunakan untuk memandu modeler melakukan penyelesaian terhadap konflik.

Penelitian selanjutnya dilakukan oleh Taentzer (Taentzer dkk, 2010) adalah melakukan pendeteksian terhadap konflik dengan menggunakan modifikasi graf. Pada penelitian tersebut dua buah model yang dibandingkan dimodelkan kedalam suatu graf. Graf tersebut mewakili perubahan yang terjadi pada setiap model. Kemudian pendeteksian konflik dilakukan dengan membandingkan graf kedua model berdasarkan tahap perubahan.

Rajbhaj (Asha Rajbhaj dan Sreedhar Reddy, 2013) melakukan penelitian yang sama dengan menggunakan graf. Penelitian ini dilakukan dengan cara menganalisa metamodel dari model yang dibandingkan. Sehingga pendeteksian konflik bisa digunakan untuk berbagai macam jenis model.

Ehrig (Ehrig dkk, 2011) melakukan penelitian untuk mencari resolusi terjadinya konflik dengan menggunakan modifikasi graf. Dalam penelitian ini metode modifikasi graf digunakan untuk mendeteksi konflik dan melakukan resolusi (menggabungkan dua buah graf yang saling konflik).

Penelitian dengan menggunakan metode modifikasi graf digunakan untuk menyelesaikan

konflik secara sintaksis, struktural dan semantik. Akan tetapi penyelesaian konflik secara semantik dengan menggunakan modifikasi graf hanya didasarkan pada persamaan perilaku disetiap tahap perubahan model. Sehingga penerapan metode modifikasi graf ini tidak bisa digunakan untuk menyelesaikan konflik semantik secara bahasa.

Penyelesaian konflik yang tidak bisa diselesaikan secara otomatis harus diselesaikan dengan mempertemukan dua belah pihak yang melakukan konflik. Brosch (Brosch dkk, 2009) membuat suatu media yang bisa mempertemukan dua belah pihak yang melakukan konflik. Media ini digunakan untuk berkomunikasi untuk menyelesaikan konflik. Akan tetapi penyelesaian konflik dengan cara ini masih membutuhkan saran dimana letak terjadinya konflik pada model.

Gomes (Gomes dkk, 2000) melakukan penelitian tentang pengembangan perangkat lunak dengan menggunakan desain perangkat lunak yang sudah ada. Gomes menggunakan ontologi pada WordNet untuk mencari kemiripan objek-objek diagram kelas antara diagram kelas lama dengan diagram kelas yang akan dibuat.

Dari studi kepustakaan yang telah dilakukan pada beberapa penelitian sebelumnya mengenai pendeteksian konflik pada model, menjadi dasar kontribusi pada penelitian ini untuk membangun sistem pendeteksian konflik dengan menggunakan metode modifikasi graf dan similaritas WordNet. Penggabungan dua metode ini dapat menyelesaikan pendeteksian konflik secara sintaksis dan semantik secara leksikal.

2. DASAR TEORI

2.1 Kategorisasi Konflik

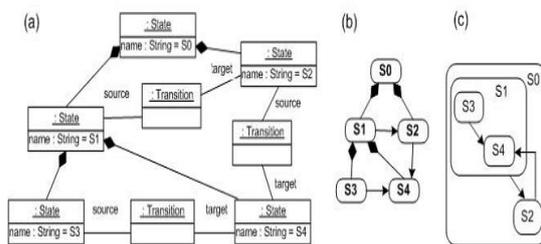
Konflik dibagi menjadi tiga kategori yaitu konflik tekstual, konflik sintaksis dan konflik semantik (Altmanninger, 2011). Konflik tekstual dapat terjadi jika perbandingan dilakukan pada representasi tekstual dari artefak. Unsur-unsur yang dibandingkan dalam pendeteksian konflik secara tekstual adalah baris teks, paragraf, kalimat, kata, atau karakter. Konflik sintaksis adalah konflik yang dapat dideteksi dengan cara membandingkan struktur dari artefak. Misalnya dua orang pengguna melakukan modifikasi nama pada elemen yang sama dan modifikasi tersebut dilakukan dengan cara yang berlawanan. Pendeteksian konflik secara sintaksis membutuhkan informasi dari metamodel, konsep bahasa dan relasi.

Konflik semantik tidak bisa dideteksi dengan menggunakan pendekatan secara sintaksis ataupun leksikal. Bahkan modifikasi pada artefak yang dilakukan secara bersama-sama akan

menghasilkan kebenaran secara sintaksis meskipun modifikasi yang dilakukan saling bertentangan. Konflik semantik dibagi menjadi 3 kategori yaitu kontradiksi statik, kontradiksi perilaku dan kontradiksi leksikal atau kontradiksi arti. Konflik semantik kontradiksi statik terjadi karena adanya pelanggaran dari aturan bahasa ketika dilakukan penggabungan pada dua buah model yang terjadi konflik. Aturan pelanggaran ini biasanya terjadi pada model, relasi, konsistensi, dan kondisi tertentu. Konflik kontradiksi perilaku adalah konflik yang terjadi karena perubahan perilaku pada model. Perilaku ini merupakan aktifitas penambahan, penghapusan dan perubahan pada suatu elemen pada model. Konflik semantik leksikal terjadi ketika model mengalami perubahan dengan bahasa yang lain tetapi memiliki arti yang sama.

2.2 Modifikasi Graf

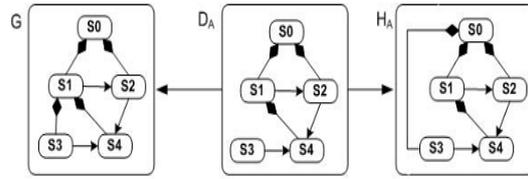
Modifikasi graf adalah pendekatan dengan mentransformasikan model dan atributnya kedalam bentuk graf. Misal pada sebuah diagram *statechart*, pada awalnya model ditransformasikan kedalam bentuk *abstract syntax* gambar gambar 1a. Dalam setiap simpul terdiri dari tipe dan atribut. Sebuah *state* dalam *statechart* diagram digambarkan menjadi sebuah simpul dengan tipe *state*. Setiap simpul mempunyai atribut S0, S1, S2, S3, S4. Sedangkan aliran aktivitas pada diagram *statechart* digambarkan dengan simpul dengan tipe transisi. Kemudian dari abstract syntax ditransformasikan kedalam *compact notation* gambar 1b. Dan kemudian ditransformasikan kedalam *concrete syntax* (graf) gambar 1c.



Gambar 1. Modifikasi Graf

Definisi dari modifikasi graf, misal dua buah graf G dan H, dengan G adalah graf sebelum mengalami perubahan dan H adalah graf setelah mengalami perubahan, dengan $G \Rightarrow H$ adalah perubahan injektif dari $G \xleftarrow{g} D \xrightarrow{h} H$. Secara berurutan $G = G_0 \Rightarrow G_1 \Rightarrow \dots \Rightarrow G_n = H$ adalah modifikasi graf secara langsung dan dinotasikan dengan $G \Rightarrow^* H$. Dimana graf D adalah graf

penengah yang berisi semua penghapusan graf yang sudah dilakukan dan tidak ada penambahan.



Gambar 2. Transformasi Graf

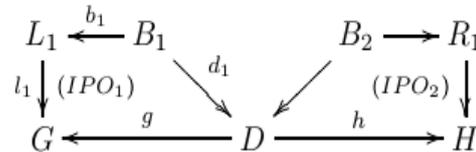
Pada gambar 2 seorang pengguna mengubah sebuah statechart diagram dari G ke HA dengan memasukkan S3 kedalam S0 dan melakukan penghapusan S3 dari S1. Graf DA adalah graf yang menunjukkan proses penghapusan S3 dari S1. Sedangkan graf HA adalah graf yang menunjukkan proses penambahan S3 kedalam S0.

2.3 Transformasi Graf

Graf yang mengalami perubahan atau transformasi dapat dicari dengan menggunakan pendekatan *double pushout* (DPO). Pendekatan DPO ini menggunakan operasi penghapusan dan penambahan pada graf. Pencarian bagian mana yang mengalami perubahan dilakukan dengan cara mencari aturan minimal. Aturan minimal adalah bagian terkecil (atom) dari sebuah graf yang mengalami perubahan.

Langkah-langkah mencari aturan minimal pada sebuah graf adalah sebagai berikut:

1. Inisialisasi *pushout*
Misal $g: D \rightarrow G$ adalah perubahan graf. *Inisial pushout* berisi perubahan graf $l_1: L_1 \rightarrow G, b_1: B_1 \rightarrow L_1$ dan $d_1: B_1 \rightarrow D$ seperti pada gambar 3.

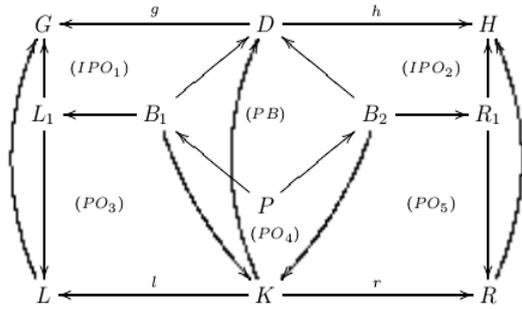


Gambar 3. Inisial Pushout

2. Mendefinisikan $B_1 \leftarrow P \rightarrow B_2$ sebagai pullback dari $B_1 \leftarrow D \rightarrow B_2$ dan $B_1 \leftarrow K \rightarrow B_2$ sebagai pushout (PO4) dari $B_1 \leftarrow P \rightarrow B_2$ dengan induksi perubahan $K \rightarrow D$
3. Bangun $B_1 \leftarrow L \rightarrow K$ sebagai *pushout* (PO3) dari $L_1 \leftarrow B_1 \rightarrow K$ dengan induksi perubahan $L \rightarrow G$. Bangun $R_1 \leftarrow R \rightarrow K$ sebagai *pushout* (PO5) dari $R_1 \leftarrow B_2 \rightarrow K$ dengan induksi perubahan $R \rightarrow H$
4. Karena IPO1 dan PO3 adalah *pushout* maka $(IPO_1) + (PO_3)$ adalah *pushout*. Dan karena IPO2 dan PO5 adalah *pushout* maka $(IPO_2) + (PO_5)$ adalah *pushout*.

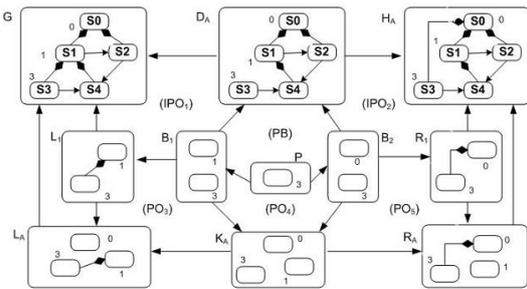
Hasil akhir konstruksi aturan minimal dapat dilihat

pada gambar 4



Gambar 4. Konstruksi Aturan Minimal

Konstruksi aturan minimal pada studi kasus gambar 2 dapat dilihat pada gambar 5.



Gambar 5. Aturan Minimal $P_A = (L_A \leftarrow K_A \rightarrow R_A)$

2.4 Karakteristik Konflik

Ehrig (Ehrig dkk, 2011), dua buah modifikasi graf m_1 dan m_2 dikatakan konflik jika memenuhi salah satu persyaratan sebagai berikut :

1. Kedua modifikasi graf tersebut menghapus elemen yang sama.
2. m_1 menghapus simpul yang akan menjadi sumber atau tujuan dari simpul baru yang ditambahkan m_2 .
3. m_2 menghapus simpul yang akan menjadi sumber atau tujuan dari simpul baru yang ditambahkan m_3 .

Jika $m_i = (G \xrightarrow{D_i} H_i)$ ($i = 1,2$), maka (m_1, m_2) dikatakan konflik jika

1. (m_1, m_2) dalam konflik delete-delete, atau
2. (m_1, m_2) dalam konflik delete-insert, atau
3. (m_2, m_1) dalam konflik delete-insert.

2.5 Similaritas Wordnet

WordNet adalah suatu sistem referensi leksikal bahasa inggris yang bersifat online. WordNet dikembangkan oleh Cognitive Science Laboratory di Universitas Princeton yang dikepalai oleh George Miller. Arti dari suatu kata pada WordNet direpresentasikan dengan synonym sets (synsets). Synsets adalah daftar term atau collocation yang artinya sama dan dalam konteks tertentu penggunaannya dapat saling

dipertukarkan. Dalam synset juga dicatat pointer-pointer ke synset lain yang digunakan untuk mendeskripsikan relasi antar synset. WordNet dibagi dalam empat taksonomi berdasarkan tipe kata yaitu kata benda, kata kerja, kata keterangan, dan kata sifat (Miller, 1993).

Wu & Palmer (Wup) menghitung similaritas dua synset dalam WordNet berdasarkan kedalaman dua synset tersebut dalam taksonomi WordNet. Pencarian similaritas antara dua synset menggunakan persamaan berikut.

$$wup_{similarity}(S_1, S_2) = \frac{2 \times depth(lcs(S_1, S_2))}{depth(S_1) + depth(S_2)} \quad (1)$$

S_1, S_2 adalah synset yang dibandingkan, $depth$ adalah kedalaman synset dalam hirarki, dan lcs (least common subsumer) adalah konsep umum.

2.5.1 Similaritas Antar Kelas

Pada penelitian ini model yang digunakan sebagai studi kasus adalah diagram kelas. Pencarian persamaan diagram kelas versi satu dengan yang lain menggunakan matrik berikut.

$$S(C_1, C_2) = \begin{bmatrix} \omega_1 \cdot S(S_1, S_2) & + \\ \omega_2 \cdot S(I_{e1}, I_{e2}) & + \\ \omega_3 \cdot S(I_{a1}, I_{a2}) & \end{bmatrix} \quad (2)$$

Dimana $S(C_1, C_2)$ adalah sebuah kelas, kelas C_1 dan kelas C_2 , $S(S_1, S_2)$ adalah tingkat kemiripan kategori dari atribut diagram kelas atau nama kelas, $S(I_{e1}, I_{e2})$ adalah tingkat kemiripan relasi antara kelas C_1 dan kelas C_2 , $S(I_{a1}, I_{a2})$ adalah tingkat kemiripan atribut dan metode antara kelas C_1 dan kelas C_2 . $\omega_1, \omega_2, \omega_3$ adalah konstanta tingkat kemiripan dimana ω_1 adalah 0,6, ω_2 adalah 0,1 dan ω_3 adalah 0,3. Nilai konstanta dari ω diperoleh dari hasil penelitian sebelumnya (Paulo dkk, 2000).

3. HASIL UJI COBA

3.1 Dataset

Dataset yang digunakan adalah kumpulan dari kelas diagram yang memiliki banyak versi. Pengumpulan dataset dilakukan dengan cara survei. Pengumpulan dataset dengan metode survei bertujuan untuk mendapatkan data dengan banyak versi dan memiliki tingkat konflik yang tinggi.

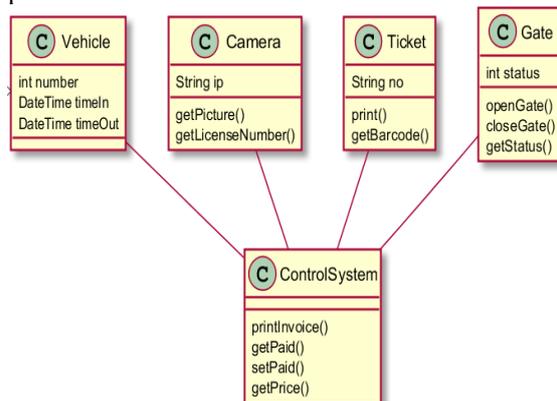
Pengumpulan dataset dengan metode survei dilakukan dengan menyebarkan angket yang berisi studi kasus. Studi kasus yang digunakan adalah narasi yang menjelaskan proses bisnis dari suatu sistem dan diagram kelas dari bisnis proses tersebut yang masih perlu penyempurnaan. Penyempurnaan kelas diagram dilakukan oleh masing-masing responden sehingga dari satu studi

SYSTEMIC

Vol. 03, No. 01, Agustus 2017, 1-8

kasus menghasilkan banyak versi dan memiliki tingkat konflik yang tinggi. Responden dari survei ini adalah dosen, mahasiswa, programmer dan analis yang bekerja diperusahaan.

Studi kasus yang digunakan adalah sistem informasi parkir mobil dengan diagram kelas pada gambar 6. Diagram kelas tersebut merupakan kelas diagram yang masih membutuhkan penyempurnaan. Dalam penelitian ini diagram kelas tersebut merupakan diagram kelas versi yang ke 0 (nol). Responden diperintahkan untuk menambahkan, mengganti ataupun menghapus kelas, nama kelas, operasi dan atribut sesuai dengan pemahaman dari penjelasan studi kasus. Pengumpulan data melalui angket tersebut menghasilkan 30 versi diagram kelas. Data ini digunakan dalam proses pengujian penelitian.



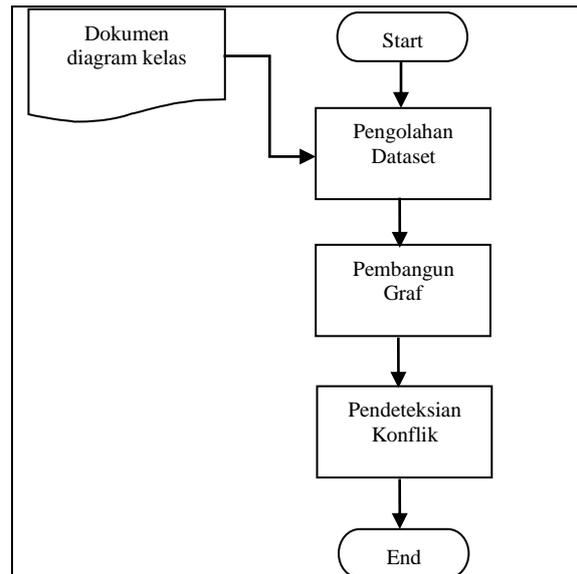
Gambar 6. Diagram Kelas Sistem Informasi Parkiran Mobil

3.2 Implementasi

Proses pendeteksian konflik pada versi kelas diagram dilakukan dengan dua tahap, proses otomatis dan proses manual. Proses pendeteksian secara manual dilakukan oleh pakar yang memahami tentang UML khususnya diagram kelas. Sedangkan untuk proses otomatis dilakukan dengan menggunakan kakas bantu. Kakas bantu dibangun dengan menggunakan bahasa pemrograman java dan pustaka pendukung sebagai berikut

1. WordNet Similarity For Java: digunakan untuk membandingkan kemiripan kata secara semantik.
2. JGraph: digunakan untuk membangun diagram kelas menjadi sebuah graf.
3. Plantuml: digunakan untuk memvisualisasikan diagram kelas.

Alur pendeteksian konflik dengan cara otomatis dapat dilihat pada gambar 7. **Error! Reference source not found.** yang meliputi proses pengolahan dataset, pembangun graf dan pendeteksian konflik



Gambar 7 Alur Kerja Pendeteksian Konflik Secara Otomatis

3.2.1 Pengolahan dataset

Dataset yang akan diujikan dalam penelitian ini diperoleh dari survei. Data hasil survei yang berisi lembaran angket atau foto angket diproses terlebih dahulu agar pendeteksian konflik bisa berjalan dengan baik. Pemrosesan ini meliputi pembenahan salah ketik, pembenahan penulisan sesuai aturan konvensi, dan konversi dari angket menjadi format plantuml.

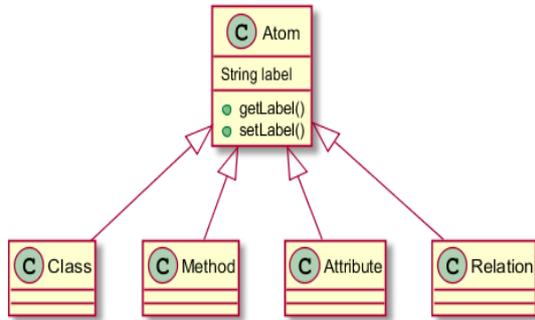
Pembenahan salah ketik dilakukan secara manual dengan melakukan pengecekan satu persatu terhadap penulisan nama komponen penyusun diagram kelas. Penulisan nama ini harus menggunakan bahasa Inggris. Jika di dalam angket yang diperoleh terdapat kata yang tidak ada dalam bahasa Inggris maka perlu ditanyakan langsung ke responden maksud dari kata tersebut.

Dalam aturan konvensi bahasa pemrograman java, nama kelas, atribut dan operasi ditulis menggunakan kata kapital (camel case). Pembenahan diagram kelas dari responden yang tidak memenuhi aturan konvensi dilakukan secara manual. Pembenahan ini dilakukan karena setiap frasa akan dibandingkan tingkat kesamaan maknanya.

Plantuml digunakan untuk memvisualisasikan diagram kelas mulai dari dataset hingga diagram kelas setelah pendeteksian konflik. Visualisasi ini digunakan untuk membandingkan secara manual apakah pendeteksian konflik sudah berjalan sesuai dengan hasil yang diharapkan.

3.2.2 Pembangun graf

Dataset yang sudah diolah menjadi format plantuml kemudian diubah menjadi bentuk graf dengan menggunakan pustaka Jgraph. Setiap komponen dalam kelas diagram akan menjadi sebuah simpul dalam graf. Simpul dalam graf merupakan objek dari kelas atom. Komponen-komponen penyusun diagram kelas, kelas, metode, atribut dan relasi merupakan instance dari kelas atom. Kelas diagram komponen penyusunnya bisa dilihat pada gambar 8.



Gambar 8 Diagram Kelas dari Objek-objek Penyusun Graf

3.2.3 Pendeteksian Konflik Dengan Modifikasi Graf

Proses pendeteksian konflik dilakukan pada diagram kelas yang sudah diproses menjadi bentuk graf. Pendeteksian dilakukan dengan mencari aturan minimal dari diagram kelas yang dibandingkan. Pendeteksian konflik menggunakan metode graf menghasilkan 3 buah jenis konflik delete-delete, insert-delete, delete-insert. Sedangkan untuk konflik insert-insert dengan metode modifikasi graf tidak diperhitungkan karena dalam metode modifikasi graf konflik insert-insert tidak dianggap sebagai konflik. Namun dalam penelitian ini konflik insert-insert dalam metode graf akan ditampilkan untuk bisa dibandingkan dengan metode similaritas WordNet.

3.2.4 Pendeteksian Konflik Dengan Similaritas Wordnet

Hasil dari pendeteksian dengan metode modifikasi graf adalah simpul dalam graf yang mengalami penambahan baru. Kemudian simpul baru ini akan dibandingkan kemiripannya dengan menggunakan similaritas WordNet. Proses pendeteksian konflik dengan similaritas WordNet menggunakan pustaka bantu WordNet similarity for java (ws4j). Penamaan kelas, atribut dan metode mengikuti aturan penulisan sesuai konvensi. Penulisan nama pada diagram kelas menggunakan format camel case. Setiap kata penyusun nama dibandingkan kemiripannya menggunakan ws4j. Untuk memecah setiap kata yang akan dibandingkan, digunakan sebuah regex.

Setiap kata yang sudah dipecah kemudian dicari nilai similaritasnya menggunakan ws4j. Nilai similaritas masing-masing kata dimasukkan ke dalam matrik nilai similaritas. Kemudian kata yang mempunyai kemiripan dipasangkan dimulai dengan nilai similaritas tertinggi ke nilai similaritas terendah. Proses pemasangan kata yang mirip ini dilakukan hingga salah satu nama memiliki kata yang sudah berpasangan semua. Kata-kata yang sudah berpasangan ini kemudian dicari rata-rata nilai similaritasnya. Nilai rata-rata similaritas yang didapatkan dibandingkan dengan nilai ambang batas. Jika nilai rata-rata similaritas lebih tinggi dari nilai ambang batas, maka nama yang dibandingkan tersebut dapat dikatakan mirip dan terjadi konflik semantik, dan sebaliknya jika nilai rata-rata similaritas lebih rendah dari nilai ambang batas, maka nama yang dibandingkan tersebut dapat dikatakan tidak mirip dan tidak terjadi konflik.

3.3 Pengujian

Pengujian dilakukan dengan membandingkan hasil identifikasi sistem dengan hasil dari pakar. Pakar yang dipilih untuk melakukan identifikasi adalah seorang yang paham dengan pengembangan perangkat lunak menggunakan metode pendekatan berorientasi objek dan memahami penggunaan UML terutama diagram kelas.

Proses identifikasi konflik dilakukan secara otomatis menggunakan kakas bantu. Kemudian hasil pengidentifikasian tersebut dibandingkan dengan pengidentifikasian secara manual oleh pakar. Tahap selanjutnya koefisien Kappa dihitung untuk melihat tingkat kecocokan antara hasil identifikasi kakas bantu dengan pakar. Deteksi konflik dilakukan dengan dua tahap, secara otomatis dan secara manual oleh pakar.

Deteksi sistem menghasilkan 991 kandidat konflik semantik. Kandidat konflik kemudian dideteksi manual oleh pakar kandidat tersebut konflik semantik atau tidak konflik. Hasil pendeteksian manual oleh pakar dapat dilihat pada tabel 1

Tabel 1 Hasil Deteksi Konflik oleh Pakar

	Konflik	Tidak Konflik
	66	925
Total Kandidat	991	

Pada skenario pengujian dilakukan beberapa percobaan terhadap studi kasus dengan rentang ambang batas similaritas 0,1 sampai 1,0. Hasil pengujian yang dilakukan oleh sistem menghasilkan kandidat yang konflik dan tidak konflik. Hasil ini kemudian dibandingkan dengan pendeteksian konflik manual oleh pakar sehingga dapat diketahui kandidat konflik yang salah dalam pendeteksian. Perbandingan hasil deteksi sistem

SYSTEMIC

Vol. 03, No. 01, Agustus 2017, 1-8

dengan hasil deteksi pakar dapat dilihat pada tabel 2.

Tabel 2 Perbandingan Hasil Deteksi Sistem

Ambang Batas Similaritas	Konflik	Tidak Konflik	Salah deteksi
0,1	991	0	925
0,2	991	0	925
0,3	829	162	763
0,4	656	335	590
0,5	526	465	460
0,6	253	738	189
0,7	127	864	89
0,8	49	942	39
0,9	32	959	38
1,0	27	964	43

Dari tabel 2, dapat dijelaskan bahwa pendeteksian konflik dengan jumlah kesalahan deteksi tertinggi yaitu sebesar 925. Kesalahan deteksi tertinggi ini terdapat pada ambang batas similaritas 0,1 dan 0,2. Tingginya jumlah kesalahan deteksi disebabkan karena pada ambang batas similaritas 0,1 dan 0,2 semua kandidat konflik terdeteksi sebagai kasus konflik semantik. Sedangkan pendeteksian dengan tingkat kesalahan terendah terdapat pada ambang batas similaritas 0,9 dengan jumlah kesalahan deteksi sebesar 38. Dari kedua hasil pada tabel 1 dan tabel 2, maka koefisien Kappa bisa dihitung dengan menggunakan persamaan 8. Hasil perhitungan koefisien kapa dengan ambang batas similaritas antara 0,1 dan 1,0 dapat dilihat pada tabel 3.

Tabel 3 Perhitungan Koefisien Kappa

Ambang Batas Similaritas	Koefisien Kappa
0,1	0,0000
0,2	0,0000
0,3	0,0275
0,4	0,0703
0,5	0,1186
0,6	0,3375
0,7	0,4945
0,8	0,6404
0,9	0,5946
1,0	0,5190

Dari tabel 3 diatas, dapat dijelaskan bahwa nilai kapa terendah terdapat pada ambang batas 0,1 dan 0,2. Sedangkan nilai kapa tertinggi terdapat pada ambang batas 0,8 yaitu 0,6404. Dari hasil koefisien Kappa tersebut maka dapat diinterpretasikan bahwa hasil tingkat kecocokan antara sistem dan pakar pada penelitian ini adalah kuat (good).

4. KESIMPULAN

Beberapa kesimpulan yang dapat ditarik dari hasil pengerjaan penelitian ini adalah sebagai

berikut.

1. Dalam penelitian ini dikembangkan suatu metode untuk mendeteksi konflik secara leksikal pada diagram kelas. Metode yang digunakan adalah modifikasi graf dan similaritas WordNet.
2. Hasil akhir uji coba oleh sistem dibandingkan dengan uji coba oleh pakar. Pendekatan ini bertujuan untuk mendapatkan koefisien Kappa dari metode yang diusulkan dalam penelitian ini. Nilai kapa tertinggi sebesar 0,6404 diperoleh ketika menggunakan nilai ambang batas similaritas 0,8.
3. Hasil perhitungan koefisien Kappa adalah 0,6404 yang kemudian dapat diinterpretasikan bahwa hasil tingkat kecocokan antara sistem dan pakar pada penelitian ini adalah kuat (good).
4. Penelitian ini menggunakan studi kasus diagram kelas sistem informasi parkir mobil. Nilai kapa yang diperoleh belum tentu sama ketika diujikan pada studi kasus yang lain.

DAFTAR PUSTAKA

- Brosch, Petra., Kargl, Horst., Seidl, Martina., Wieland, Konrad., Wimmer, Manuel., dan Kappel, Gerti.,(2011), "Conflict as First-Class Entities: A UML Profile for Model Versioning", MODELS 2010 Workshop, Springer-Verlag, Berlin Heidelberg.
- Miller, George A., Beckwith, Richard., Fellbaum, Christiane., Gross, Derek., dan Miller, Katherine., (1993), "Introduction to WordNet: An On-line Lexical Database"
- Altmanninger, K., dan Pierantonio, A., (2011), "A categorization for conflicts in model versioning", Elektrotechnik & Informationstechnik, Springer
- Ehrig, Hartmut., Ermel, Claudia., dan Taentzer, Gabriele. (2011), "A Formal Resolution Strategy for Operation-Based Conflicts in Model Versioning Using Graph Modifications".
- Rajbhoj, Asha., dan Reddy, Sreedhar., (2013), "A Graph-Pattern Based Approach for Meta-Model Specific Conflict Detection in a General-Purpose Model Versioning System"
- Taentzer, Gabriele., Ermel, Claudia., Langer, Philip., dan Wimmer., Manuel, (2010), "Conflict Detection for Model Versioning Based on Graph Modifications"

- Gomes, Paulo., C Pereira, Francisco., Paiva, Paulo., Seco, Nuno., Carreiro, Paulo., L. Ferreira, José., dan Bento, Carlos., (2000), "Case Retrieval of Software Designs using WordNet"
- Brosch, Petra., Seidl, Martina., Wieland, Konrad., dan Wimmer, Manuel., (2009), "We can work it out: Collaborative Conflict Resolution in Model Versioning"
- Altman, DG, (1991), "Practical Statistics for Medical Research"
- Sim, J., & Wright, C. C., (2005), "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements", *Physical Therapy*, Vol. 85(3), hal. 257–68.