

Rancang Bangun Kakas Bantu Deteksi Ketidaksesuaian Kode Sumber terhadap Diagram Urutan

Albert Bungaran Manik dan Daniel Oranova Siahaan

Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: danielos@cs.its.ac.id

Abstrak—Pada daur hidup perangkat lunak sendiri terdapat beberapa tahapan-tahapan yang harus dilalui yaitu pengumpulan kebutuhan, desain, implementasi, testing, perawatan. Dalam fokus studi ini, studi dititik beratkan pada tahap terakhir daur hidup perangkat lunak, yaitu tahap perawatan perangkat lunak. Dalam daur hidup tahap perawatan sangat rentan untuk mengubah kode sumber sesuai dengan kebutuhan atau keperluan fitur pada sebuah perangkat lunak yang mengakibatkan tidak sesuai kode sumber terhadap desain awal. Tujuan studi ini yaitu untuk membuat sebuah kakas bantu untuk membantu analis sistem dalam mendeteksi ketidaksesuaian kode sumber tersebut terhadap desain, dalam hal ini desain yang menjadi pembanding adalah diagram urutan, hal ini dikarenakan diagram urutan menggambarkan alur sebuah fitur atau sistem. Dalam studi ini kakas bantu akan dibentuk menggunakan bahasa pemrograman java. Diagram urutan dengan format xmi dan kode sumber dengan format xml merupakan input yang diperlukan untuk kakas bantu. Dalam studi ini pendeteksian ketidaksesuaian dilakukan menggunakan string matching dan word similarity. Kakas bantu ini nanti menghasilkan berupa sebuah alur dari diagram urutan yang ditunjukkan pada tanda benar dan salah. Setiap alur digambarkan menjadi sebuah triplet. Triplet meliputi subyek, predikat, dan obyek. Dalam program ini nanti akan diuji dengan data uji sebanyak 10 buah dataset yang berasal dari 5 buah kasus diagram urutan. Hasil program menunjukkan 95% kesepakatan terhadap analisis ahli rekayasa perangkat lunak.

Kata Kunci—Sequence Diagram, Kode Sumber, Deteksi Diagram Urutan, Ketidak Sesuaian.

I. PENDAHULUAN

Pada sebuah pengembangan perangkat lunak, terdapat daur hidup yang membagi tahapan aktifitas pengembangan perangkat lunak. Tahapan aktivitas tersebut meliputi tahap perancangan, tahap analisis, tahap desain, tahap pembuatan, tahap pengujian, dan tahap perawatan perangkat lunak. Fokus dari penelitian proposal ini terletak pada tahapan perawatan perangkat lunak. Tahap ini merupakan tahap yang sangat rentan untuk merubah kode sumber untuk bisa memenuhi kebutuhan pengguna. Tentu perubahan kode sumber kedepannya pasti mempengaruhi perangkat lunak dari desain awal perangkat lunak tersebut. Pada kasus penelitian ini, desain yang digunakan sebagai pengukur ketidaksesuaian adalah diagram urutan. Diagram urutan dipilih menjadi pengukur karena diagram urutan menggambarkan bagaimana sebuah alur fitur atau sistem berinteraksi. Dari alur tersebut nantinya akan dideteksi ketidaksesuaian dari kode sumber.

Dari penelitian terkait yang dilakukan oleh rujukan [7] dilakukan sebuah pendeteksian untuk mendeteksi perubahan kode sumber yang telah dimodifikasi untuk mendeteksi *plagiarism* (plagiat) dengan menggunakan *similarity detection running karp rabin* yang menunjukkan hasil deteksi *plagiarism* dari perubahan kode sumber.

Dalam studi ini dilakukan penelitian yang merujuk dari penelitian terkait [7] dimana penelitian melakukan pendeteksian ketidaksesuaian diagram urutan. Studi ini nanti akan melakukan pendeteksian ketidaksesuaian kode sumber terhadap diagram urutan yang diharapkan bisa menghasilkan sebuah hasil yang menunjukkan ketidaksesuaian yang terdapat dalam kode sumber.

II. TINJAUAN PUSTAKA

A. XMI

XML Metadata Interchange (XMI) adalah sebuah standar pertukaran informasi *metadata* melalui *Extensible Markup Language* (XML). XMI menggabungkan sebuah *meta-meta-model* ke sebuah teks dalam format XML. Untuk menghasilkan informasi *metadata* ke sebuah teks dalam format XML, XMI mendefinisikan serangkaian aturan *meta-model* berdasarkan MOF *meta-meta-model*. Salah satu contoh *meta-model* adalah UML. Sebuah UML didefinisikan sebagai sebuah model yang terdiri dari informasi *metadata* yang saling berkaitan karena mendeskripsikan informasi yang saling berhubungan, mendefinisikan sebuah rangkaian aturan yang mendefinisikan struktur dan konsistensi, dan memiliki makna dalam sebuah kerangka.

B. WordNet

WordNet adalah sebuah basis data leksikal dalam bahasa inggris. *WordNet* ini nantinya mendeskripsikan makna suatu kata dan mengelompokkan kata sesuai abjad. Kata yang berdekatan memiliki makna yang berhubungan. Dalam kasus studi ini nantinya setiap kata atau kalimat yang ditemukan akan dihitung kemiripannya. Perhitungan kemiripan tersebut dilakukan dengan menggunakan *Wu-Palmer*. Perhitungan *Wu-Palmer* ditunjukkan pada persamaan (1) dibawah ini

$$Sim_{wup}(c1, c2) = \frac{2 \times depth(LCS(c1, c2))}{depth(c1) + depth(c2)} \quad (1)$$

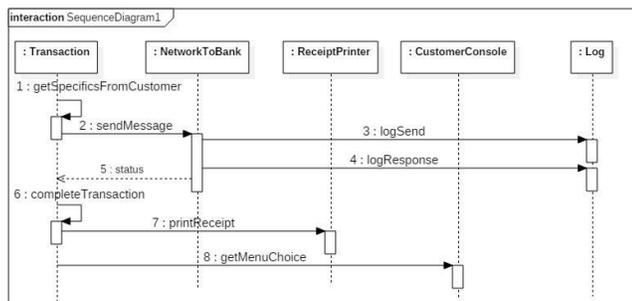
dari setiap hasil dari persamaan tersebut akan dilakukan *metric similarity*. Dari hasil tersebut nantinya akan menjadi hasil yang menentukan apakah kedua kalimat tersebut berhubungan.

C. String Matching

Metode *string matching* digunakan untuk melakukan pencarian semua kemunculan string. Pada studi ini Algoritma string matching yang digunakan adalah fungsi *equals* pada pustaka yang telah disediakan oleh java.. Metode ini digunakan untuk mendeteksi persamaan dari dua buah string yang ada,yang berguna untuk menemukan kesamaan antara dua buah string.

D. Ketidaksesuaian Kode Sumber

Ketidaksesuaian antara diagram urutan (*sequence diagram*) dan kode sumber bisa dilihat dari ada atau tidak nya nama metode dan kelas yang di cocokan berdasarkan diagram urutan yang sudah dibuat. Dengan mengetahui terdapat kecocokan metode dan kelas antara kode sumber dan diagram urutan Kita dapat mengetahui ketidaksesuaian antara kode sumber yang dibentuk dengan perancangan sistem melalui diagram urutan (*sequence diagram*). Contoh kasus ketidak sesuaian diagram urutan dan kode sumber adalah sebagai berikut.



Gambar 1. Contoh kasus diagram urutan.

Salah satu contoh kasus diagram urutan transaksi pada ATM (*Automatic Teller Machine*), Dari Gambar 1 diagram urutan menunjukkan bagaimana setiap *object* dari *transaction.java* berinteraksi satu sama lain. Contoh kode sumber untuk fungsi *transaction* yang dimana *sequence diagram* tidak sesuai dengan kode sumber di gambarkan pada Gambar 2.

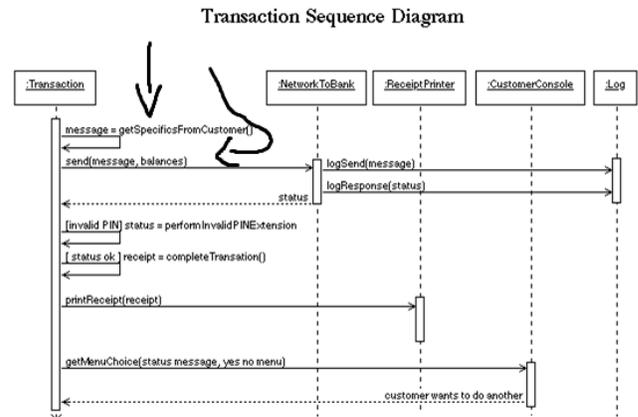
```

Public boolean Transaction() throws CardRetained
{
    .....
    while (true){
        switch(state)
        {
            .....
            try
            {
                atm.getCustomerConsole().display("");
                state = SENDING_TO_BANK_STATE;
            }
            catch(CustomerConsole.Cancelled e)
            {
                .....
            }
            case SENDING_TO_BANK_STATE:
            {
                .....
            }
            else
            {
                state = ASKING_DO_ANOTHER_STATE;
            }
        }
        break;
    }
}
    
```

Gambar 2. Ilustrasi Potongan kode transaction.

Dari Gambar 2 didapat kode sumber tidak sesuai dengan diagram urutan yang diberikan,karena kode sumber tidak

memiliki sebuah metode *getSpecificsFromCustomer()* dan metode `status = atm.getNetworkToBank().sendMessage(message, balances).` Karena dalam *Transaction* tidak terdapat 2 metode tersebut maka diagram urutan dikategorikan tidak sesuai dengan kode sumber yang ada. Berikut hasil setelah pendeteksian ketidak sesuaian diagram urutan dapat diilustrasikan seperti Gambar 3



Gambar 3. Contoh hasil analisis.

Gambar tersebut menunjukkan Alur kode sumber pada `message = getSpecificsFromCustomer()` tidak sesuai dengan diagram urutan, dan alur kode sumber `send(message,balances)` juga tidak sesuai terhadap diagram urutan.

E. StarUML

Star Uml adalah sebuah tools *Unified Modeling Language (UML)* yang di kembangkan oleh *MKLab*. *StarUML* sendiri merupakan sebuah aplikasi berbasis dekstop yang digunakan untuk membuat diagram *Unified Modeling Language*. Dalam kaskas bantu ini nantinya diagram urutan akan dibentuk menjadi *xmi*.

F. Xmltranslator

XMLtranslator merupakan sebuah program berbasis web yang memetakan sebuah kode sumber berbasis object ke dalam sebuah *AST (Abstract Syntax Tree)*. Hasil dari *xmltranslator* berupa file *xml*. Dalam aplikasi ini nantinya kode sumber akan dibentuk menjadi *xml*.

III. METODE PENELITIAN

A. Deskripsi Umum Sistem

Studi ini berbentuk perangkat lunak untuk mendeteksi Ketidaksesuaian kode sumber terhadap Diagram urutan. Perangkat lunak berupa kaskas bantu berbasis dekstop. Adapun pengguna dari kaskas bantu ini nantinya adalah analis sistem,yang bertanggung jawab terhadap perubahan kode sumber terhadap diagram urutan.

Gambaran proses dari kaskas bantu yang dibuat adalah sebagai berikut :

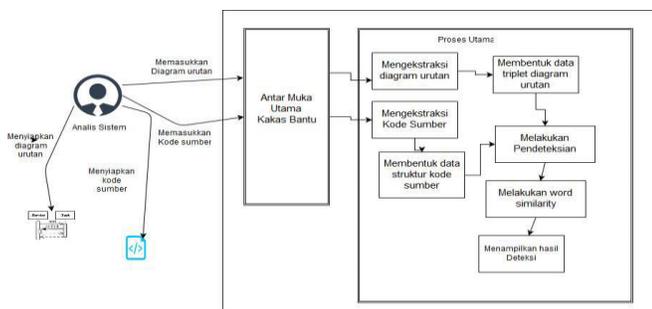
1. Pra-Proses: Analisis sistem memilih diagram urutan yang akan dipakai sebagai acuan sebagai pendeteksian ketidaksesuaian, analisis memilih diagram urutan dengan menggunakan aplikasi *Star-UML*, kemudian Analisis sistem mengonversikan Diagram Urutan tersebut menjadi sebuah berkas dalam format "XMI" menggunakan *extension Star*

UML 2. Berkas XMI yang dihasilkan selanjutnya Analisis Memilih Kode sumber yang berkaitan terhadap diagram urutan kemudian analisis sistem mengkonversikan Kode Sumber menjadi berkas dalam format “XMI” menggunakan *xmltranslatorApp* [1]. Berkas yang dihasilkan XMI kode sumber yang dihasilkan nantinya disebut dengan kumpulan dokumen kode sumber.

2. *Input*: diagram urutan yang memiliki format “.xmi”, dan dokumen kode sumber yang berisi beberapa kode sumber berkaitan dengan diagram urutan dalam format “.xmi”.
3. *Proses*: mengekstraksi diagram urutan yang terdiri dari *lifeline*, *message*, *fragment*, *ownedAttribute* dan *packaged element*. dari hasil ekstraksi diagram urutan diolah untuk mendapatkan triplet yang membentuk subyek, predikat, dan obyek. Kemudian mengekstraksi folder yang terdiri banyak kode sumber, dari masing-masing file kode sumber dilakukan ekstraksi untuk mengambil *nama kelas* dari kode sumber, lalu *nama metode* yang terdapat di dalam kelas tersebut, kemudian *baris kode sumber* yang berisi dari pemanggilan metode didalam kelas obyek. Kemudian setelah mendapatkan ekstraksi dari diagram urutan dan kode sumber dilakukan proses matching yang dilakukan untuk mendeteksi setiap triplet dengan melakukan pendeteksian didalam kode sumber. Nantinya hasil dari proses ini akan membentuk sebuah penanda pada triplet, nantinya penanda itu akan menandakan kesesuaian kode sumber terhadap diagram urutan. Triplet yang benar akan mendapatkan tanda centang (✓) pada subyek, predikat, dan obyeknya. Untuk triplet tidak sesuai akan mendapatkan tanda silang (✗) pada subyek, predikat, dan obyeknya.
4. *Output*: Berupa daftar triplet yang berisi subyek, predikat dan obyek dengan penanda centang (✓) atau penanda silang (✗) pada subyek, predikat, dan obyek.

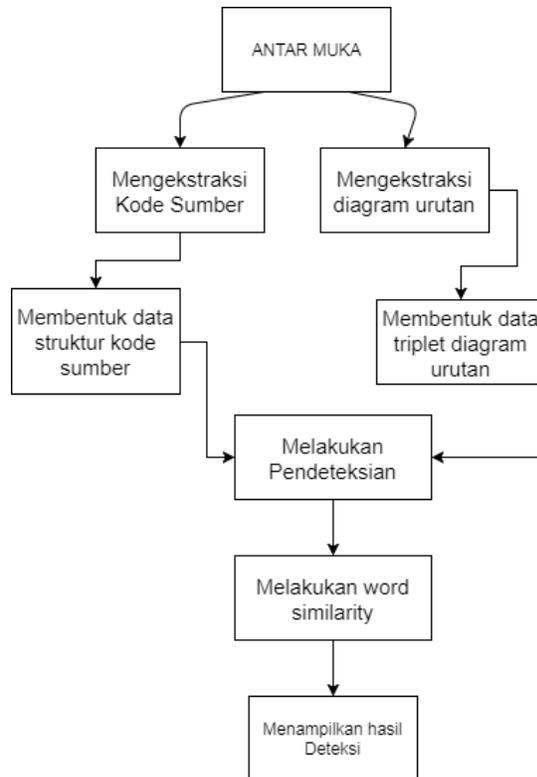
B. Perancangan Proses Kakas Bantu

Pendeteksian ketidaksesuaian diagram urutan merupakan Proses yang terjadi di dalam kakas bantu yang hendak diimplementasikan. Implementasi tersebut membahas bagaimana alur dan metode untuk mendeteksi ketidaksesuaian antara kode sumber dengan diagram urutan. Secara garis besar kakas bantu ini memiliki 4 proses utama yaitu proses untuk mengekstraksi, proses pendeteksian, proses word similarity, dan proses menampilkan hasil pendeteksian. Proses utama tersebut digambarkan melalui arsitektur perangkat lunak pada Gambar 4



Gambar 4. Gambar arsitektur kakas bantu.

Dari hasil arsitektur diatas didapati fungsi proses utama ditunjukkan pada Gambar 5



Gambar 5. Fungsi utama kakas bantu.

1) Menyiapkan Kode Sumber dan Diagram Urutan

Penyiapan kode sumber dan diagram urutan dilakukan dengan memilih diagram urutan dan kode sumber yang telah sesuai aturan, aturan yang dimaksud adalah aturan dalam penamaan dan alur pembuatan baik diagram urutan dan kode sumber. Aturan diagram urutan adalah pembuatan diagram urutan harus dimulai dengan pembuatan kelas diagram lebih dahulu. Untuk kode sumber aturannya adalah penamaan kode sumber harus berupa *camel-Case*. Lalu setelah itu apabila sudah dipersiapkan sedemikian rupa, masing-masing diagram urutan dan kode sumber diekspor dengan menggunakan StarUML 2 untuk diagram urutan, dan *xmltranslator* untuk kode sumber. Fase ini digambarkan sebagai antarmuka didalam Gambar 5.

2) Mengekstraksi Diagram Urutan dan Kode Sumber

Dari hasil ekspor yang dilakukan sebelumnya, dilanjutkan dengan mengekstraksi elemen-elemen xml yang telah diekspor sebelumnya. Kedua ekstraksi ini menggunakan *SAXparser* dimana *SAXparser* ini adalah pustaka pembaca elemen xml. Untuk masing-masing ekstraksi diagram urutan dan kode sumber pendekatan metode untuk ekstraksi dilakukan secara berbeda. Untuk diagram urutan metode yang digunakan adalah metode *flag* dimana setiap hasil elemen tag awal diketahui dan akhir tag diketahui maka dilakukan proses ekstraksi. Untuk kode sumber ekstraksi dilakukan menggunakan metode *stack* dimana setiap elemen tag xml yang ditemukan disimpan ke dalam stack dengan tujuan untuk bisa memilah tag-tag yang ada di dalam xml kode sumber.

Tabel 1.
Hasil pengujian 10 dataset

Penga mat	Kasus									
	1	2	3	4	5	6	7	8	9	10
Kakas Bantu - Ahli	0.3	1.0	1.0	1.0	1.0	1.0	0.8	1.0	0.3	1.0

3) Membentuk Data Struktur Diagram Urutan dan Kode Sumber

Setelah berhasil menyimpan hasil ekstraksi dari proses sebelumnya. Hasil ekstraksi tersebut dibentuk menjadi sebuah data struktur yang dapat memudahkan untuk melakukan pendeteksian. Dalam hal ini diagram urutan dirubah menjadi sebuah triplet dimana triplet berisikan subyek, predikat, dan obyek. Untuk kode sumber dilakukan pembentukan data struktur menjadi nama kelas, isi metode dalam kelas, dan pemanggilan kode sumber dalam kelas.

4) Melakukan Pendeteksian

Setelah berhasil membentuk data struktur baik diagram urutan maupun kode sumber, selanjutnya dilakukan pendeteksian ketidaksesuaian. Pendeteksian dilakukan dengan menggunakan string matching dari pustaka java. Dilakukan string matching dengan mencari subyek dengan kriteria subyek merupakan kelas, dan subyek terdapat pemanggilan kode sumber metode dalam kelas lain. Selanjutnya dilanjutkan dengan pendeteksian predikat, predikat dideteksi dengan kriteria predikat merupakan sebuah metode dalam kelas obyek. Sedangkan untuk obyek memiliki kriteria adalah sebuah kelas. Setelah itu dilakukan pendeteksian pada setiap triplet dari diagram urutan. Triplet tersebut menggambarkan alur dari diagram urutan yang telah diekstraksi oleh kakas bantu ini.

5) Melakukan Word Similarity

Word similarity dimulai dengan mendapatkan hasil dari elemen `ArrayList` yang telah didapatkan pada pembentukan data struktur kode sumber dan elemen `ArrayList` triplet diagram urutan. Setiap pendeteksian *word similarity* dilakukan untuk mendeteksi kembali predikat dengan cara mendeteksi masing-masing nama metode. Pendeteksian word similarity dilakukan pertama dengan mendeteksi predikat dengan nama metode pada kode sumber. Deteksi dilakukan pertama kali dengan melakukan *pre processing* dengan merubah suatu penamaan metode menjadi sebuah kata dan begitu sebaliknya pada predikat yang terdapat pada triplet juga. Nantinya dari kata itu dihitung jarak perbedaan makna antara kata dari predikat dan kata dari nama kelas kode sumber, apabila didapati hasil melebihi dari threshold maka didapati predikat diberi tanda benar

IV. PENGUJIAN

Pengujian dilakukan dengan skenario melakukan pengujian hasil pendeteksian dengan membandingkan hasil kakas bantu dengan hasil analisis ahli rekayasa perangkat lunak, hasil perbandingan ditunjukkan dengan nilai kesepakatan *kappa statistic*. Dari hasil pengujian yang dilakukan antara hasil ketidaksesuaian kakas bantu dan ahli didapati hasil pengujian terhadap 10 buah dataset dari 5 buah fitur diagram urutan. Hasil pengujian ditunjukkan pada Tabel 1.

Dari Tabel 1 apabila dihitung secara keseluruhan nilai kesepakatan total antara ketidaksesuaian ahli dan hasil dari kakas bantu didapati nilai sejumlah 0.9 yang menunjukkan kesepakatan ahli bahwa kakas bantu dapat mendeteksi ketidaksesuaian.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Kakas Bantu yang dibuat berhasil memenuhi tujuan utamanya untuk mendeteksi ketidaksesuaian kode sumber terhadap diagram urutan.
2. Berdasarkan hasil dari kappa statistic yang didapatkan, hasil deteksi ketidaksesuaian yang dihasilkan oleh kakas bantu menghasilkan kesepakatan dengan ahli sejumlah 0.95. nilai ini menunjukkan kesepakatan antara kakas bantu dan ahli rekayasa perangkat lunak yang termasuk dalam *almost perfect*
3. Berdasarkan evaluasi non-fungsional dari hasil uji coba yang dilakukan partisipan menyetujui bahwa kakas bantu ini mudah digunakan, tetapi agak kurang mudah apabila tidak didampingi dengan orang lain
4. Dari pengerjaan kakas bantu didapat ekstraksi diagram urutan dilakukan dengan menggunakan SAX sebagai pembaca xmi dan pengambilan elemennya menggunakan metode Flag, dimana menandai setiap elemen yang ditemukan. Sedangkan untuk kode sumber didapati ekstraksi diagram urutan menggunakan SAX sebagai pembaca xml dan pengambilan elemennya menggunakan metode stack dimana setiap elemen tag xml yang ditemukan disimpan ke dalam stack.

DAFTAR PUSTAKA

- [1] XML translator, "PHP to XML Converter," *xmltranslator.appspot.com*. [Online]. Available: xmltranslator.appspot.com/sourcecodetoxml.html.