

RMTool; Sebuah Aplikasi Pemodelan Persyaratan Perangkat Lunak menggunakan UML

Egia Rosi Subhiyako¹, Danang Wahyu Utomo²

Abstract—Requirement Engineering is one of stages in the software development. One of the challenges in requirements engineering is the use of tools in modeling the requirements. The number of requirements modeling applications does not guarantee that an analyst can model the needs appropriately for the complexity of the application, because the cost of modeling applications is too expensive to affect the use of such applications to model the requirements. Another problem faced by students on education is the difficulty in applying knowledge about software engineering. For example, if students are given a project from the industry, the requirements elicitation and modeling application are usually needed. The tools used in the industry are sometimes not suitable for education purposes. This paper aims to develop a tool for requirements modeling, to assist novice analyst in learning and classifying UML diagrams as well as learning the syntax. The tool is named RMTool, which is an object-oriented modeling tool based on web. Testing is conducted both on the developer and the user environment. Evaluation by end users involves 76 students as novice analyst to test the effectiveness and usefulness of modeling tool using questionnaire. The evaluation results show that the majority of students are satisfied using the application.

Intisari—Rekayasa persyaratan atau kebutuhan adalah salah satu tahapan dalam pembangunan perangkat lunak. Salah satu tantangan dalam rekayasa kebutuhan adalah penggunaan aplikasi bantu dalam pemodelan kebutuhan. Banyaknya aplikasi pemodelan kebutuhan tidak menjamin seorang analis dapat memodelkan kebutuhan dengan tepat guna dikarenakan kompleksnya aplikasi kebutuhan yang ada. Relatif mahalnnya aplikasi pemodelan juga memengaruhi penggunaan aplikasi tersebut untuk memodelkan kebutuhan. Masalah lain yang dihadapi siswa dalam dunia pendidikan adalah kesulitan menerapkan pengetahuan tentang rekayasa perangkat lunak dalam dunia nyata. Sebagai contoh, apabila siswa diberikan proyek dari dunia industri, biasanya siswa akan melakukan elisitasi kebutuhan dan memerlukan aplikasi pemodelan. Aplikasi yang digunakan di dunia industri kadangkala tidak cocok untuk digunakan dalam dunia pendidikan. Makalah ini bertujuan merancang perangkat lunak pemodelan kebutuhan guna membantu analis pemula dalam mempelajari dan mengklasifikasikan diagram UML serta mempelajari *syntax* dan memodelkan kebutuhan. Aplikasi yang dibangun bernama RMTool, yang merupakan aplikasi pemodelan kebutuhan berorientasi objek berbasis *web*. Setelah RMTool dibangun, dilakukan pengujian, baik di lingkungan pengembang maupun di lingkungan pengguna. Evaluasi RMTool di lingkungan pengguna melibatkan 76 siswa sebagai analis pemula untuk menguji efektifitas dan kegunaan aplikasi pemodelan menggunakan kuesioner. Berdasarkan hasil evaluasi dapat

disimpulkan bahwa mayoritas siswa puas menggunakan aplikasi tersebut.

Kata Kunci— RMTool, Rekayasa Persyaratan, Perangkat Lunak, UML, Pemula

I. PENDAHULUAN

Proses pengembangan perangkat lunak mengikuti beberapa tahapan yang bertujuan untuk mendapatkan perangkat lunak yang berkualitas. Untuk menjadi seorang pengembang perangkat lunak yang memiliki kompetensi, sudah sewajarnya seseorang mempunyai kemampuan yang mencakup semua tahap pengembangan [1]. Pada prinsipnya, dalam pengembangan perangkat lunak terdapat dua prinsip, yakni pendidikan di universitas dan pendidikan di industri [2]. Keduanya memiliki tujuan dan karakteristik yang berbeda dalam level usia, pengalaman, latar belakang, dan kemampuan. Dalam pendidikan, siswa harus memiliki pengetahuan dan kemampuan yang mencakup semua tahapan. Dalam pembangunan perangkat lunak, siswa diharuskan memodelkan masalah dalam tahap analisis dan memodelkan solusi dalam tahap perancangan. Pemodelan perangkat lunak menggunakan *Unified Modeling Language* (UML) dinyatakan sebagai standar *de facto* pemodelan berorientasi objek [3]. Pemodelan persyaratan berguna untuk mengomunikasikan dengan semua pihak yang berkepentingan dan sebagai sebuah *blueprint*. Di antara aplikasi-aplikasi pemodelan persyaratan komersial adalah Enterprise Architect, Rational Rose, MagicDrawUML, dan PowerDesigner. Sedangkan aplikasi pemodelan persyaratan yang gratis adalah StarUML, ArgoUML, dan UMLDesigner. Siswa biasanya menggunakan aplikasi pemodelan seperti Rational Rose atau ArgoUML [4]. Pemodelan persyaratan lebih fokus kepada pengembangan profesional daripada pengajaran [5]. Aplikasi pemodelan persyaratan yang tersedia relatif membingungkan dan relatif sulit untuk digunakan serta aplikasi pemodelan persyaratan yang cocok untuk pembelajaran memiliki harga yang mahal [6]. Pemaparan bahwa permasalahan dalam dunia pendidikan adalah budaya kerja pengembangan sistem dan metode formal dalam pengajaran rekayasa perangkat lunak [7].

Makalah ini bertujuan untuk membangun aplikasi pemodelan persyaratan perangkat lunak menggunakan UML, atau secara khusus, untuk membangun aplikasi pemodelan persyaratan yang cocok untuk proses pembelajaran dalam dunia pendidikan yang dibangun berbasis *web* menggunakan bahasa pemrograman JSP, dengan MySQL sebagai basis data. Pengembangan sebuah sistem harus mengikuti tahapan pengembangan yang seharusnya. Penilaian dalam pengembangan sistem dipaparkan menjadi hal yang penting demi kelancaran pengembangan sistem tersebut [8]. Penelitian

^{1,2}Dosen, Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Jl. Imam Bonjol 207 Semarang, 50131, Indonesia (telp: 024-3517261; fax: 024-3569684; e-mail: egia@dsn.dinus.ac.id)

terkait dengan paper ini yakni penemuan banyak notasi yang biasanya dibutuhkan untuk menggambarkan dan memodelkan dalam pengembangan sistem [9]. Banyak kondisi aplikasi pemodelan yang tersedia hanya mendukung satu atau beberapa notasi. Hal tersebut menjadi pekerjaan seorang analis untuk menganalisis dan memilih aplikasi yang tepat untuk memodelkan. Penelitian lain memaparkan aplikasi Papyrus Tool [10]. Aplikasi ini memiliki keuntungan, yakni pengetahuan mengenai persyaratan dapat diklasifikasikan dalam model UML, dan hal ini dapat mendukung analisis otomatis. Dalam pemodelan persyaratan seperti UML, digunakan UMLet [11]. UMLet merupakan aplikasi yang menggunakan bahasa pemrograman Java dan mendukung beberapa platform seperti Windows, OS X, dan Linux. UMLet apabila digunakan dalam lingkungan pendidikan maupun industri. Penelitian lain memaparkan penggunaan aplikasi pemodelan dengan responden analis bisnis, *programmer*, dan *tester* [12]. Studi yang dilakukan bertujuan untuk melakukan evaluasi efektivitas aplikasi-aplikasi UML berdasarkan pandangan dan pendapat dari setiap responden. Aplikasi-aplikasi yang dibandingkan adalah Visual Paradigm, Enterprise Architect, Metamill, ArgoUML, dan UML Diagrammer. Kustomisasi aplikasi pemodelan persyaratan juga pernah dilakukan dalam membangun *System Modeling Language* (sysML) yang bertujuan untuk menganalisis, spesifikasi, perancangan, dan juga verifikasi sistem yang membingungkan [13]. Alasan kustomisasi aplikasi pemodelan persyaratan juga tersedia di lingkungan industri. Hal ini disebabkan dunia industri tidak memiliki aplikasi yang cocok untuk kerangka kerja [14]. Studi ini menggunakan pendekatan baru untuk mendukung strategi yang digunakan.

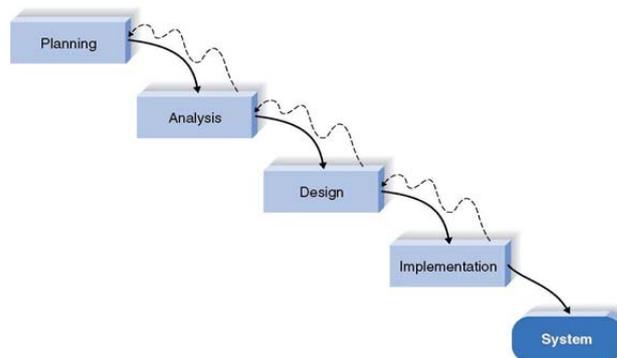
II. METODOLOGI

Metode pengembangan sistem yang digunakan adalah *waterfall* model. Gbr. 1 menunjukkan fase-fase pengembangan sistem dengan *waterfall* model. Model ini memiliki karakteristik seperti menghasilkan dokumen di setiap tahapnya, tidak ada tahap yang dilakukan bersama-sama, dan setiap tahap berpengaruh ke tahap selanjutnya. Berdasarkan Gbr. 1, tahap awal dalam model ini adalah perencanaan pengembangan sistem, kemudian analisis yaitu memodelkan masalah ke dalam spesifikasi persyaratan, perancangan yang berisi cara memodelkan solusi yang telah dipilih, implementasi yaitu tahap transformasi dari perancangan ke *coding*, sampai akhirnya didapatkan sebuah sistem [15].

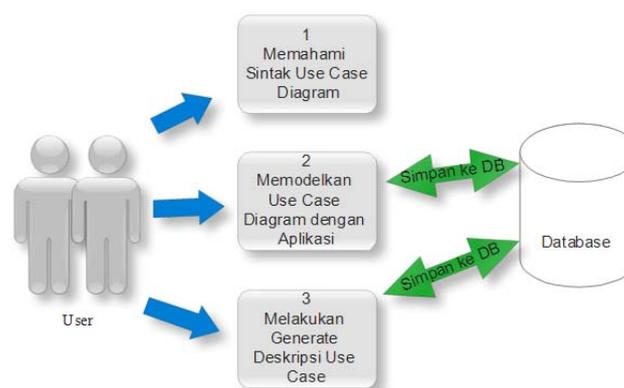
III. ANALISIS DAN PERANCANGAN

Pengembangan aplikasi meliputi beberapa tahapan, yakni rekayasa kebutuhan, analisis, perancangan, implementasi, dan pengujian [16]. Dilakukannya *improve* dalam sebuah aplikasi juga merupakan hal yang penting [17]. Aplikasi pemodelan dibangun berdasarkan permasalahan sebelumnya. Motivasi pembangunan aplikasi adalah tersedianya aplikasi yang mudah dipahami dan mendukung pemodelan persyaratan dalam pendidikan. Tujuan dari pengembangan aplikasi ini adalah membantu analis pemula dalam memahami *syntax* dalam diagram UML, khususnya diagram *use case* yang

menggambarkan fungsionalitas sebuah sistem. Aplikasi ini dikembangkan menggunakan bahasa pemrograman JSP, HTML, dan Javascript. Untuk basis data digunakan MySQL dan Apache Tomcat sebagai *web server*. Untuk *editor* digunakan Eclipse Juno dan SQLYog untuk membuat basis data. Aplikasi pemodelan persyaratan yang dibangun sebelumnya dilakukan analisis dan perancangan dengan metode berorientasi objek. Dalam metode berorientasi objek digunakan pemodelan dengan diagram-diagram UML seperti diagram *use case*, diagram kelas, dan diagram aktivitas. Dalam makalah ini aplikasi digambarkan dengan diagram *use case* untuk fungsionalitas aplikasi, diagram kelas untuk menggambarkan kelas dan hubungannya, serta diagram aktivitas untuk mengetahui alur kegiatan dalam sistem.



Gbr. 1 Model *waterfall* [15].

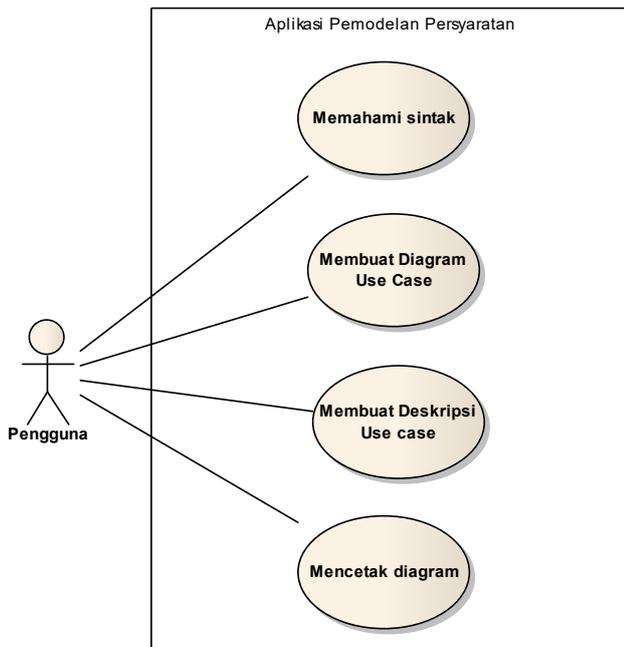


Gbr. 2 Arsitektur aplikasi.

Gbr. 2 menunjukkan arsitektur aplikasi yang dibangun. Arsitektur aplikasi menggambarkan hubungan antara aplikasi yang digunakan oleh pengguna untuk terhubung ke aplikasi. Terdapat menu-menu utama, di antaranya adalah pengenalan *syntax-syntax*, penggunaan diagram *use case* dengan studi kasus, penjelasan deskripsi *use case*, dan tentang aplikasi. Dari hasil *generate* diagram *use case* dan deskripsi *use case* diperoleh keluaran berupa diagram yang sudah dibuat.

A. Diagram Use Case

Diagram *use case* merupakan inti dalam analisis dan perancangan aplikasi berorientasi objek. Diagram *use case* menunjukkan fungsi-fungsi atau fitur utama yang ada dalam sebuah aplikasi.



Gbr. 3 Diagram use case aplikasi.

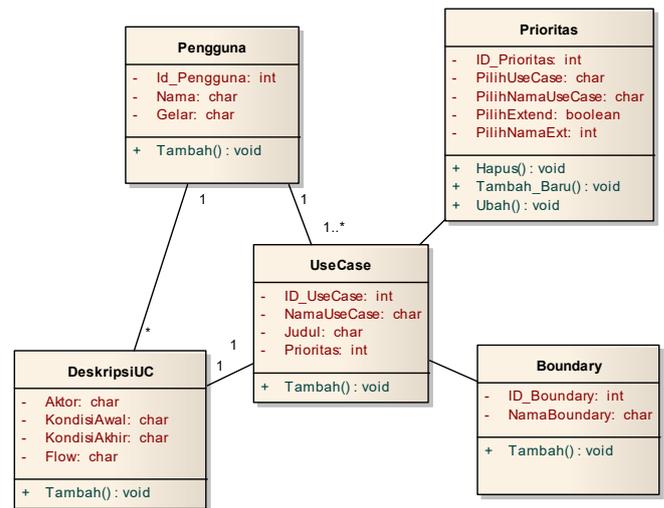
Gbr. 3 adalah diagram *use case* yang menggambarkan fungsionalitas aplikasi dan menunjukkan hubungan antara aktor dan *use case* yang terkait. Dalam diagram *use case* aplikasi tersebut terdapat satu aktor yakni pengguna aplikasi. Pengguna dapat memahami *syntax* terlebih dahulu, kemudian membuat diagram *use case* sesuai dengan studi kasus yang diberikan. Pembuatan diagram *use case* ini menggunakan konsep *drag and drop*. Lalu, pengguna dapat membuat deskripsi *use case* atau berupa skenario sesuai dengan *use case* yang telah dibuat. Selanjutnya, pengguna dapat mencetak diagram hasil yang telah dibuat.

B. Diagram Kelas

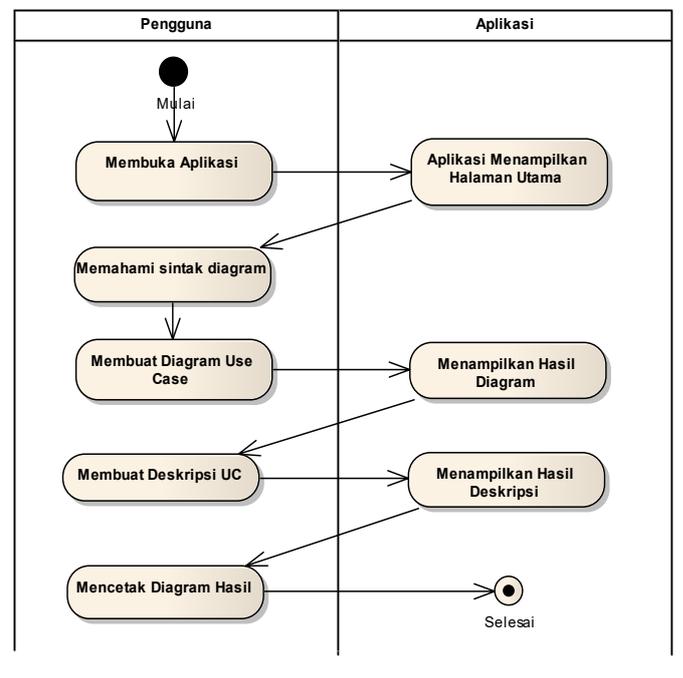
Diagram kelas dalam sebuah aplikasi menggambarkan jenis objek dalam aplikasi serta hubungan statis yang terjadi. Di dalam diagram kelas juga ditunjukkan *attribute* dan *method* yang dilakukan. Gbr. 4 menunjukkan diagram kelas yang ada dalam aplikasi yang akan dibangun. Terdapat lima kelas yang ada dalam aplikasi ini. Kelas pertama adalah pengguna yang mempunyai relasi dengan kelas *use case* dan kelas deskripsi *use case*. Kemudian, terdapat kelas prioritas yang berfungsi untuk mengendalikan prioritas dalam fungsi *drag and drop* ketika membuat diagram *use case*. Kelas terakhir adalah kelas *boundary* yang merupakan pembatasan sistem atau aplikasi di dalam sebuah diagram *use case*.

C. Diagram Aktivitas

Diagram aktivitas berfungsi untuk menunjukkan urutan aktivitas yang terjadi di dalam aplikasi yang dibangun. Diagram aktivitas yang digambarkan merupakan aktivitas *end-to-end* aplikasi. Aktivitas pertama yang dilakukan adalah pengguna memulai membuka aplikasi, kemudian aplikasi akan memberikan timbal balik. Gbr. 5 menjelaskan tahapan aktivitas yang dilakukan dalam aplikasi yang dibangun.



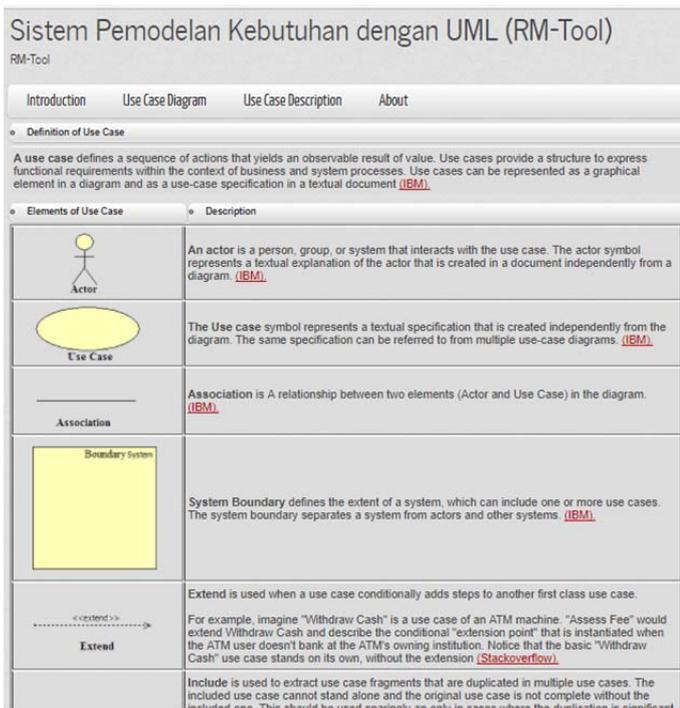
Gbr. 4 Diagram kelas aplikasi.



Gbr. 5 Diagram aktivitas aplikasi.

IV. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil implementasi dan pengujian aplikasi yang telah dilakukan. Implementasi dilakukan menggunakan bahasa pemrograman JSP dan MySQL sebagai basis data. Strategi dan teknik pengujian yang digunakan adalah yang populer atau paling sering digunakan, seperti pengujian menggunakan metode *blackbox* maupun *whitebox* dan atau strategi *top down* maupun *bottom up* [16]. Gbr. 6 menunjukkan pendahuluan aplikasi yang dibangun. Dalam Gbr. 6 dijelaskan *syntax-syntax* yang ada dalam diagram *use case*. Terdapat dua bagian, yakni menu yang mendefinisikan simbol-simbol utama, dan bagian kanan adalah definisi dan fungsi setiap *syntax* dalam diagram *use case*.



Gbr. 6 Tampilan halaman utama.



Gbr. 7 Tampilan halaman membuat diagram use case.

Gbr. 7 adalah komponen utama dari aplikasi pemodelan persyaratan yang dibangun. Terdapat dua bagian utama, yaitu masukan pengguna dan keluaran yang dihasilkan. Pada bagian ini dijelaskan cara melakukan pembuatan diagram *use case* dengan aplikasi ini. Cara kerja aplikasi ini adalah sebagai berikut.

1. Pengguna memberikan masukan dalam *boundary name* yang berisi batasan/ruang lingkup sistem. Pemberian nama ini hanya satu kali.
2. Pengguna memberikan masukan berupa nama aktor, yang dapat dilakukan berulang kali disesuaikan dengan jumlah aktor dalam sebuah studi kasus.

3. Pengguna memasukkan nama *use case*, yang dapat dilakukan beberapa kali sesuai jumlah *use case* dalam studi kasus.
4. Pengguna juga dapat memberikan fungsi *include* ataupun *extend* terhadap *use case* dalam sebuah sistem.
5. Keluaran sistem akan ditampilkan pada bagian kanan halaman tersebut.



Gbr. 8 Tampilan halaman membuat deskripsi use case.

TABEL I
PENGUJIAN FUNGSI PEMBUATAN DIAGRAM *USE CASE*

Kasus Data dan Hasil Uji (Data Benar)	
Data masukan	Pembuatan diagram <i>use case</i> (data diisi dengan benar)
Yang diharapkan	Data tersimpan dan menampilkan model <i>use case</i> yang benar
Pengamatan	Data tersimpan dan menampilkan model <i>use case</i> yang sesuai dengan masukan
Kesimpulan	Pengujian diterima
Kasus Data dan Hasil Uji (Data Salah)	
Data masukan	Pembuatan diagram <i>use case</i> diisi data kosong
Yang diharapkan	Muncul pesan <i>javascript</i> bahwa harus diisi
Pengamatan	Muncul pesan kesalahan
Kesimpulan	Pengujian diterima

Gbr. 8 menunjukkan pembuatan deskripsi *use case*. Di sini, pengguna hanya memberikan masukan dan deskripsi *use case* akan ditampilkan pada bagian kanan.

Setelah tahapan implementasi dilakukan, tahap selanjutnya adalah melakukan pengujian secara *alpha* dan *beta*. Pengujian *alpha* dilakukan di lingkungan pengembang aplikasi. Terdapat beberapa teknik pengujian yang dapat dilakukan, salah satunya adalah teknik pengujian *black box* [16]. Pengujian ini juga dilakukan untuk menguji fungsionalitas [18], [19].

TABEL II
PENGUJIAN FUNGSI PEMBUATAN DESKRIPSI *USE CASE*

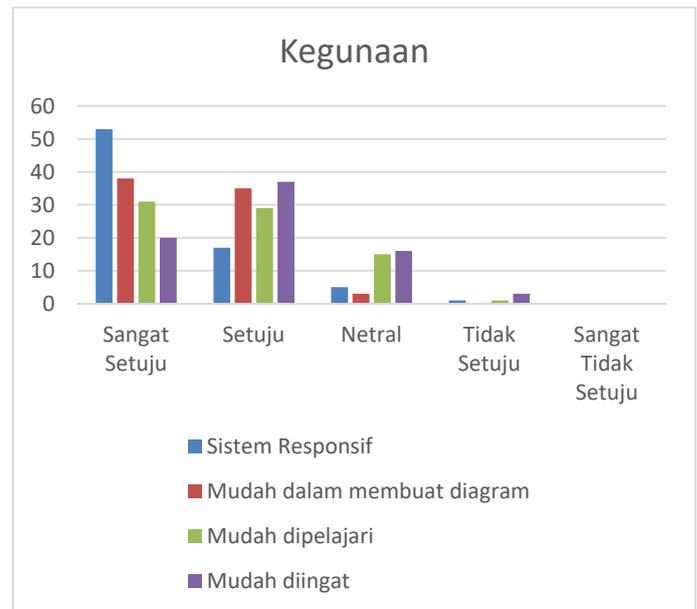
Kasus Data dan Hasil Uji (Data Benar)	
Data masukan	Aktor, kondisi awal, kondisi akhir, dan <i>flow</i>
Yang diharapkan	Pesan “Data tersimpan” muncul
Pengamatan	Jika tombol “ <i>submit and display</i> ” ditekan data tersimpan
Kesimpulan	Pengujian diterima
Kasus Data dan Hasil Uji (Data Salah)	
Data masukan	Aktor, kondisi awal, kondisi akhir, dan <i>flow</i>
Yang diharapkan	Muncul pesan “data tidak boleh kosong”
Pengamatan	Muncul pesan kesalahan
Kesimpulan	Pengujian diterima

Pengujian menggunakan metode *black box* telah dilakukan dan ditampilkan pada Tabel I dan Tabel II. Fungsi aplikasi telah berjalan sesuai dengan yang diharapkan. Setelah pengujian di lingkungan pengembang aplikasi dilakukan, selanjutnya dilakukan pengujian di level *end user* atau pengguna akhir.

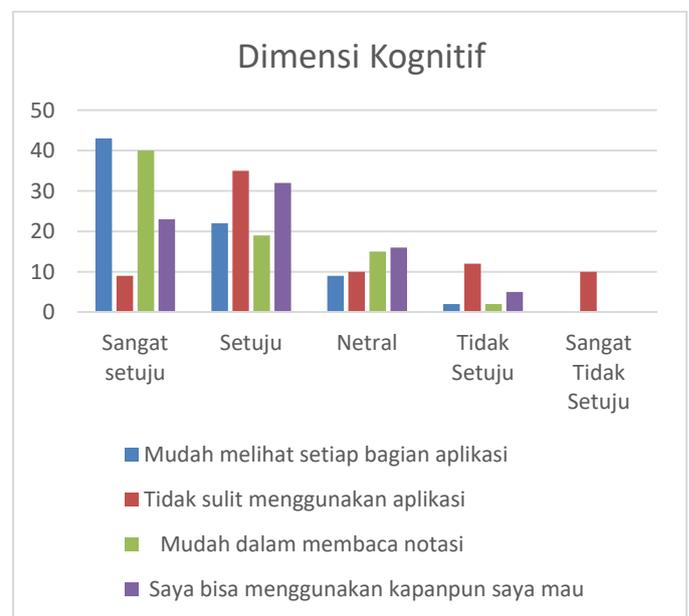
Pada bagian ini disajikan hasil evaluasi yang didapatkan dengan mengevaluasi pengguna akhir mengenai persepsi mereka terhadap aplikasi ini. Evaluasi digunakan untuk mengukur penggunaan, dimensi kognitif, dan kepuasan pengguna. Parameter kegunaan, kemudahan penggunaan, dan kemudahan dipelajari berpengaruh kepada kepuasan pengguna dalam menggunakan sistem yang dibangun [20]. Tahapan pengujian tidak menjamin kualitas sistem, tetapi dapat memberikan tingkat kepercayaan terhadap aplikasi tersebut [21]. Pengguna yang berpartisipasi dalam studi ini adalah 76 orang responden siswa yang mengambil mata kuliah rekayasa perangkat lunak, yang terdiri atas 59 laki-laki dan 17 perempuan. Setiap responden diberi penjelasan singkat mengenai aplikasi yang dibangun. Kemudian kepada responden diberikan waktu untuk mengeksplorasi dan menyelesaikan sebuah studi kasus dengan memodelkan persyaratan dengan aplikasi tersebut. Terdapat sebelas pertanyaan yang diberikan, yang mencakup tingkat respons aplikasi, kemudahan pembuatan diagram, kemudahan pembelajaran, kemudahan mengingat aplikasi, kerja keras dalam penggunaan aplikasi, penjelasan notasi, kepuasan pengguna, sampai dengan pemberian rekomendasi kepada teman responden untuk menggunakan aplikasi tersebut.

Berdasarkan hasil evaluasi kuesioner yang diberikan akan dilakukan analisis menggunakan skala Likert 1 sampai 5, yakni sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju.

Persentase diperoleh dengan rumus $Y = \frac{P}{Q} \times 100\%$, dengan Y adalah nilai persentase, P adalah jumlah responden yang menjawab pertanyaan, dan Q adalah jumlah responden.



Gbr. 9 Evaluasi kegunaan.

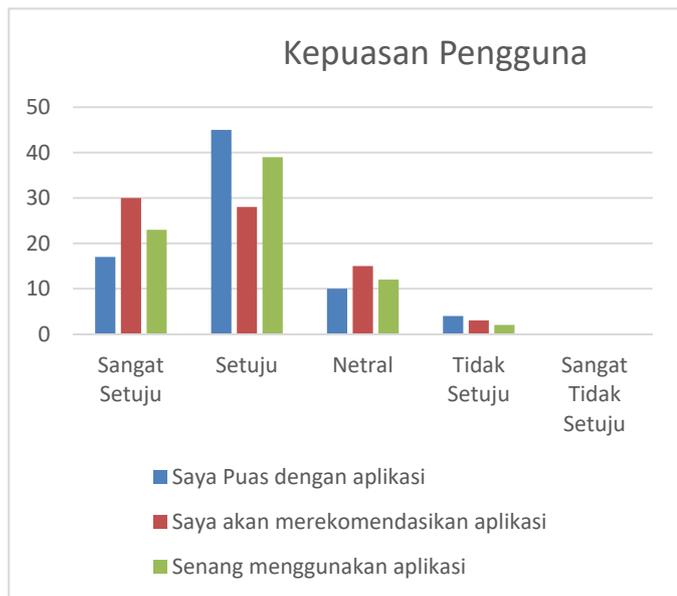


Gbr. 10 Evaluasi dimensi kognitif.

Hasil evaluasi terbagi menjadi tiga bagian utama, yakni evaluasi kegunaan, evaluasi dimensi kognitif, dan evaluasi kepuasan. Gbr. 9 menunjukkan hasil survei tentang kegunaan. Berdasarkan grafik yang ditunjukkan, diperoleh hasil positif bahwa sistem responsif 92%, mudah dalam penggunaan 96%, mudah dipelajari 79%, dan mudah diingat 75%. Hal tersebut menunjukkan bahwa responden mayoritas sangat setuju dan setuju bahwa aplikasi berguna karena responsif, mudah dalam membuat diagram, mudah dipelajari, dan mudah untuk diingat.

Gbr. 10 menunjukkan evaluasi berdasarkan dimensi kognitif. Hasil evaluasi menunjukkan bahwa 85% responden sangat setuju dan setuju bahwa mudah untuk melihat setiap bagian dari aplikasi, 57% responden sangat setuju dan setuju bahwa tidak sulit menggunakan aplikasi tersebut, 78%

responden sangat setuju dan setuju bahwa mudah untuk membaca setiap bagian notasi dalam aplikasi, dan 72% responden dapat menggunakan aplikasi RMTTool kapanpun responden membutuhkan.



Gbr. 11 Evaluasi kepuasan pengguna.

TABEL III
PERBANDINGAN APLIKASI PEMODELAN PERSYARATAN PERANGKAT LUNAK

Aplikasi	Diagram use case	Teknologi (Bahasa)			Cocok untuk		Biaya	
		Java	PHP	JSP	Industri	Pendidikan	Gratis	Berbayar
Rational Rose	√	√			√			√
Enterprise Architect	√	√	√		√			√
MS. Visio	√					√		√
Magic Draw	√	√			√			√
UMLet	√	√				√	√	
STORM	√					√	√	
RMTTool	√	√	√	√		√	√	

Gbr. 11 menunjukkan evaluasi berdasarkan kepuasan pengguna. Hasil evaluasi menunjukkan bahwa 62 responden, atau 82% menyatakan puas dengan aplikasi ini, 76% akan merekomendasikan aplikasi ini, dan 82% senang menggunakan aplikasi ini. Dari hal tersebut dapat disimpulkan bahwa RMTTool dapat digunakan sebagai solusi alternatif dalam kaitannya sebagai aplikasi pemodelan persyaratan perangkat lunak menggunakan diagram UML.

Perbandingan aplikasi pemodelan persyaratan perangkat lunak sebelumnya yang membandingkan beberapa aplikasi yakni Rational Rose, Argouml, MagicDraw, dan Enterprise Architect [11], [22]. Disebutkan bahwa Rational Rose merupakan pilihan terbaik dalam mendukung semua aktivitas

dalam sistem, kemudian ArgoUML merupakan pilihan dalam perancangan dan arsitektur sistem, MagicDraw merupakan pilihan terbaik untuk analisis dan menulis dokumentasi, sedangkan Enterprise Architect adalah pilihan terbaik dalam model bisnis.

Tabel III menunjukkan perbandingan aplikasi pemodelan kebutuhan perangkat lunak berdasarkan teknologi yang digunakan, cocok untuk digunakan baik dalam dunia pendidikan maupun dunia industri, dan biaya, baik gratis maupun berbayar. Aplikasi yang dibandingkan yakni Rational Rose, Enterprise Architect, MS. Visio, MagicDraw, UMLet, STORM, serta RMTTool yang merupakan aplikasi yang dibangun. Hasil perbandingan menunjukkan bahwa aplikasi yang dibangun yakni RMTTool mendukung pembuatan diagram use case, yang didukung oleh bahasa pemrograman Java, PHP, dan JSP. Selain itu, RMTTool juga cocok untuk dunia pendidikan dan gratis. Berdasarkan hasil perbandingan, RMTTool dapat dijadikan sebagai alternatif solusi berdasarkan masalah dalam makalah ini, yakni aplikasi gratis dan cocok untuk dunia pendidikan, khususnya untuk analis pemula yang sedang belajar memahami *syntax-syntax* dalam diagram UML.

V. KESIMPULAN

Proses rekayasa persyaratan perangkat lunak tidak dapat dipisahkan dari memodelkan persyaratan. Untuk melakukan pemodelan persyaratan, analis menggunakan aplikasi pemodelan seperti Enterprise Architect, Rational Rose, ArgoUML, dan StarUML. Aplikasi pemodelan kebutuhan yang tersedia relatif mahal, kompleks, dan tidak cocok untuk digunakan dalam dunia pendidikan. Siswa terkadang hanya mamahami sedikit dan bahkan tidak memahami fungsi dan penggunaan *syntax-syntax* dalam diagram UML. Solusi dari masalah tersebut adalah mengembangkan aplikasi pemodelan persyaratan yang cocok untuk pendidikan dan mudah dipahami oleh siswa atau seorang analis pemula. Dalam makalah ini, telah dibangun sebuah aplikasi bernama RMTTool. RMTTool dikembangkan berbasis *web* dengan JSP sebagai bahasa pemrograman dan MySQL sebagai basis datanya. Aplikasi ini ditujukan untuk pemodelan persyaratan perangkat lunak di dunia pendidikan. Evaluasi telah dilakukan dengan melihat beberapa aspek, seperti aspek kegunaan, aspek dimensi kognitif, dan aspek kepuasan pengguna. Evaluasi melibatkan 76 responden dengan beberapa latar belakang seperti siswa dan analis pemula. Responden diberi kuesioner setelah sebelumnya mendapatkan pelatihan singkat penggunaan aplikasi ini. Hasil evaluasi aplikasi menunjukkan nilai positif untuk setiap aspek dan mayoritas pengguna puas menggunakan aplikasi ini. Diperkuat lagi dengan hasil perbandingan aplikasi, yaitu RMTTool merupakan aplikasi gratis dan cocok untuk dunia pendidikan, khususnya untuk analis pemula yang belajar memahami *syntax-syntax* dalam diagram UML. Studi selanjutnya adalah melakukan perbaikan dan penambahan fitur serta menambah diagram UML lain.

UCAPAN TERIMA KASIH

Terima kasih ditujukan untuk DP2M-DIKTI yang telah membiayai Penelitian Dosen Pemula (PDP) yang dibebankan

kepada Daftar Isian Pelaksanaan Anggaran (DIPA) Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi dan Pendidikan Tinggi Nomor SP DIPA-042.06.1.401516/2017 dan nomor kontrak 085/A.38.04/UDN-09/IV/2017, pada 28 April 2017 di LPPM Universitas Dian Nuswantoro Semarang.

REFERENSI

- [1] E. R. Subhiyacto, M. Kamalrudin, S. Sidek, and S. S. S. Ahmad, "Customization of Requirements Modelling Tools for Software Engineering Education," *Int. Symp. Res. Innov. Sustain. 2014*, vol. 26, no. 4, pp. 1581–1584, 2014.
- [2] H. D. Mills, "Software engineering education," *Proc. IEEE*, vol. 68, no. 9, pp. 1158–1162, 1980.
- [3] X. He, "Formalizing UML Class Diagrams - A Hierarchical Predicate Transition Net Approach," *24th Annu. Int. Comput. Softw. Appl. Conf.*, 2000.
- [4] Wendy Boggs, *Mastering UML with rational rose 2002*, SYBEX Samp. 2002.
- [5] K. Alfert and J. Schr, "Software Engineering Education Needs Adequate Modeling Tools," *Proceedings of the 17th Conference on Software Engineering Education and Training (CSEET'04)*, 2004, pp. 1–6.
- [6] J. M. Carrillo de Gea, J. Nicolás, J. L. Fernández Alemán, A. Toval, C. Ebert, and A. Vizcaíno, "Requirements engineering tools: Capabilities , survey and assessment," *Inf. Softw. Technol.*, vol. 54, no. 10, pp. 1142–1157, Oct. 2012.
- [7] B. Malik and S. Zafar, "A Systematic Mapping Study on Software Engineering Education," *World Acad. Sci. Eng. Technol.*, vol. 7, no. 1, pp. 2061–2071, 2012.
- [8] D. W. Utomo and E. R. Subhiyacto, "Assessing Novice Teams in Collaborative Software Engineering Education," *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, 2016, pp. 23–28.
- [9] S. Supakkul and L. Chung, "The RE-Tools: A multi-notational requirements modeling toolkit," *2012 20th IEEE International Requirements Engineering Conference, RE 2012 - Proceedings*, 2012, pp. 333–334.
- [10] H. Dubois, F. Lakhali, and S. Gérard, "The Papyrus tool as an Eclipse UML2-modeling environment for requirements," *2009 2nd International Workshop on Managing Requirements Knowledge, MARK 2009*, 2009, pp. 27–31.
- [11] M. Auer, T. Tschurtschenthaler, and S. Biffl, "A flyweight UML modelling tool for software development in heterogeneous environments," *Proc. 20th IEEE Instrum. Technol. Conf. (Cat No 03CH37412) EURMIC-03*, pp. 267–272, 2003.
- [12] A. Bobkowska and M. Weihs, "Verification of the fit to user profiles for UML tools," *Proceedings of the 2008 1st International Conference on Information Technology, IT 2008*, 2008, no. May.
- [13] M. Hause and F. Thom, "Modeling High Level Requirements in UML/SysML," *{INCOSE} 2005 {UK} Spring Conf.*, pp. 10–15, 2005.
- [14] D. Silingas and R. Butleris, "Towards customizing UML tools for enterprise architecture modeling," *Inf. Syst.*, pp. 1–8, 2009.
- [15] A. Dennis, B. H. Wixom, and R. M. Roth, *System Analysis and Design*. 2012.
- [16] E. R. Subhiyacto and D. W. Utomo, "Software Testing Techniques And Strategies Use In Novice Software Teams," *SISFO*, vol. 5, no. 5, 2016.
- [17] D. W. Utomo, E. R. Subhiyacto, and S. Ahmad, "Tool Enhancement For Collaborative Software Engineering Education," *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA 2015)*, 2015, pp. 9–16.
- [18] E. N. Jannah and A. Hidayah, "Sistem Terintegrasi Berbasis Web untuk Pencarian dan Pemesanan Kelompok Seni Pertunjukan," *JNTETI*, vol. 5, no. 4, pp. 245–251, 2016.
- [19] R. Yunis and K. Telaumbanua, "Pengembangan E-Learning Berbasis LMS untuk Sekolah, Studi Kasus SMA/SMK di Sumatera Utara," *JNTETI*, vol. 6, no. 1, pp. 32–36, 2017.
- [20] W. A. Kusuma, V. Noviasari, and G. I. Marthasari, "Analisis Usability dalam User Experience pada Sistem KRS- Online UMM menggunakan USE Questionnaire," *JNTETI*, vol. 5, no. 4, pp. 294–301, 2016.
- [21] E. R. Subhiyacto and D. W. Utomo, "Strategi, teknik, faktor pendukung dan penghambat pengujian untuk pengembang perangkat lunak pemula," *SENTIKA*, pp. 236–241, 2016.
- [22] L. Khaled, "A comparison between UML tools," *2nd International Conference on Environmental and Computer Science, ICECS 2009*, 2009, pp. 111–114.