

ANALISIS DAN PERANCANGAN APLIKASI PEMODELAN KEBUTUHAN PERANGKAT LUNAK MENGGUNAKAN METODE PROTOTYPING

Egia Rosi Subhiyacto¹, Danang Wahyu Utomo²

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang
Jl Imam Bonjol No 207 Semarang
Telp. (024) 3517261
E-mail: egia@dsn.dinus.ac.id

ABSTRAK

Pengembangan sebuah perangkat lunak tidak terlepas dari peran seorang analis. Salah satu tugas analis adalah memodelkan masalah dalam tahapan analisis. Dalam memodelkan, analis biasanya menggunakan alat bantu berupa aplikasi pemodelan kebutuhan perangkat lunak. Aplikasi pemodelan yang tersedia di pasaran relatif kompleks dan mahal. Hal tersebut menyebabkan analis mengalami masalah dalam memodelkan kebutuhan, khususnya untuk seorang analis pemula. Analis pemula biasanya kesulitan menerapkan pengetahuan mereka apabila aplikasi yang digunakan terlalu kompleks dan sulit dipahami. Aplikasi yang tersedia juga lebih banyak ditujukan untuk dunia industri, dan jarang sekali aplikasi yang ditujukan untuk pendidikan. Dalam paper ini bertujuan untuk menganalisis dan merancang sebuah aplikasi pemodelan kebutuhan perangkat lunak. Aplikasi yang dirancang ditujukan untuk dunia pendidikan dan dimaksudkan membantu analis pemula dalam memahami sintak-sintak khususnya dalam diagram UML (*Unified Modeling Language*). Metode pengembangan sistem yang digunakan adalah metode prototyping. Penggunaan metode ini adalah untuk mendapatkan umpan balik yang cepat dari pengguna, membantu analis pemula dalam menentukan kebutuhan pengguna yang sebenarnya, dan evaluasi dapat dilakukan berkali-kali. Penelitian ini menghasilkan usulan hasil analisis dan perancangan aplikasi pemodelan kebutuhan perangkat lunak berorientasi objek.

Kata Kunci: Analisis, Perancangan, Pemodelan, Kebutuhan, Perangkat Lunak

11. PENDAHULUAN

Bagian ini membahas tentang latar belakang penelitian, motivasi, masalah dan tujuan.

Latar Belakang

Prinsip pengembangan sebuah perangkat lunak terbagi menjadi dua yaitu untuk tujuan pendidikan dan untuk tujuan industri (Mills, 1980). Terdapat perbedaan yang mendasar dari kedua prinsip tersebut jika dilihat dari segi pengalaman, level usia, kemampuan maupun latar belakang. Khusus untuk tujuan pendidikan perangkat lunak sudah seharusnya seorang pengembang memiliki kemampuan yang meliputi semua tahapan (Subhiyacto, Kamalrudin, Sidek, & S. Ahmad, 2014). Dalam dunia pendidikan perangkat lunak terdapat beberapa tahapan pengembangan yang lebih dikenal dengan istilah SDLC (*Software Development Life Cycle*), yang mencakup tahap perencanaan, analisis, perancangan, dan implementasi. Lebih fokus pada tahap analisis, ketika seorang pengembang melakukan proses analisis, maka dilakukan tahapan pemodelan menggunakan berbagai macam diagram. Pemodelan kebutuhan perangkat lunak yang dilakukan terbagi menjadi dua jenis, yakni pemodelan terstruktur dan pemodelan berorientasi objek. Perbedaan mendasar dari pemodelan tersebut dapat dilihat dari perangkat lunak yang dibangun. Pemodelan terstruktur, digunakan untuk perangkat lunak yang menggunakan modul-modul, sedangkan pemodelan berorientasi objek didasarkan pada *class* dan objek yang ada pada sebuah perangkat lunak. Jika pada pemodelan terstruktur menggunakan DFD (*Data Flow Diagram*) untuk menggambarkan prosesnya dan ERD (*Entity Relationship Diagram*) untuk menggambarkan relasi datanya, maka untuk pemodelan berorientasi objek menggunakan pemodelan dengan UML (*Unified Modeling Language*). UML telah menjadi standar pemodelan berorientasi objek (He, 2000). Dalam UML terdapat beberapa diagram, yang familiar diantaranya adalah *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*. Sedangkan untuk melakukan pemodelan itu sendiri, pengembang biasanya memerlukan alat bantu berupa aplikasi pemodelan seperti *Rational Rose*, *StarUML*, *ArgoUML*, *Enterprise Architect*, dan berbagai alat pemodelan lainnya.

Motivasi dan Tujuan

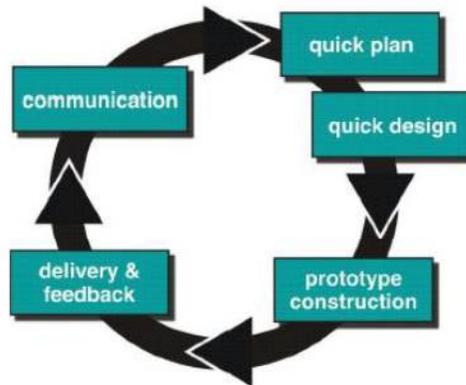
Aplikasi pemodelan perangkat lunak yang tersedia mayoritas ditujukan untuk tujuan industri (Alfert & Schr, 2004). Hal tersebut berbanding lurus dengan kenyataan bahwa aplikasi yang tersedia relative mahal dan kompleks apabila digunakan dalam dunia pendidikan atau pengajaran (Carrillo de Gea et al., 2012). Makalah ini memiliki tujuan melakukan analisis dan perancangan aplikasi pemodelan kebutuhan perangkat lunak berorientasi objek. Secara khusus memiliki tujuan untuk membantu dalam dunia pengejaran dalam hal pemodelan kebutuhan perangkat lunak. Aplikasi yang dirancang dimaksudkan membantu pengembang perangkat lunak pemula dalam memahami sintak ataupun simbol-simbol diagram dalam UML dan kegunaannya.

12. PENELITIAN TERKAIT

Penelitian sebelumnya telah dilakukan dalam paper (Khaled, 2009) yang melakukan perbandingan beberapa alat pemodelan kebutuhan seperti *Rational Rose*, *ArgoUML*, *MagicDraw*, dan *Enterprise Architect*. Perbandingan dilakukan untuk memilih yang cocok dalam mendukung aktivitas dalam perancangan arsitektur sistem. Menulis dokumentasi, dan pemodelan proses bisnis. Dalam (Supakkul & Chung, 2012) ditemukan banyak sintak atau notasi yang digunakan dan diperlukan untuk melakukan pemodelan dalam pengembangan sistem perangkat lunak. Terdapat beberapa kondisi aplikasi yang hanya mendukung satu atau sedikit notasi. Hal tersebut memungkinkan analisis dalam memilih aplikasi yang tepat dalam memodelkan kebutuhan. Penelitian lain dalam (Auer, Tschurtschenthaler, & Biffi, 2003) menggunakan aplikasi UMLet yang merupakan aplikasi pemodelan kebutuhan perangkat lunak berorientasi objek. Aplikasi tersebut cocok digunakan di dunia pendidikan maupun industri. Aplikasi pemodelan yang dikustomisasi juga terdapat pada (David & Idasiak, 2009) yang membangun aplikasi bernama sysML yang mempunyai tujuan dalam proses analisis, spesifikasi, merancang dan memverifikasi sistem yang membingungkan. Terkait dengan dunia industri, dalam (Silingas, 2008) dilakukan penelitian mengenai pengembangan aplikasi yang terkait dengan kerangka kerja di dunia industri, dengan menggunakan pendekatan baru yang mendukung strategi tersebut.

13. METODOLOGI

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode prototyping. Penggunaan metode ini didasarkan pada keuntungan dalam mendapatkan umpan balik yang cepat dari *customer*.



Gambar 1. Model prototyping (Pressman & Maxim, 2014)

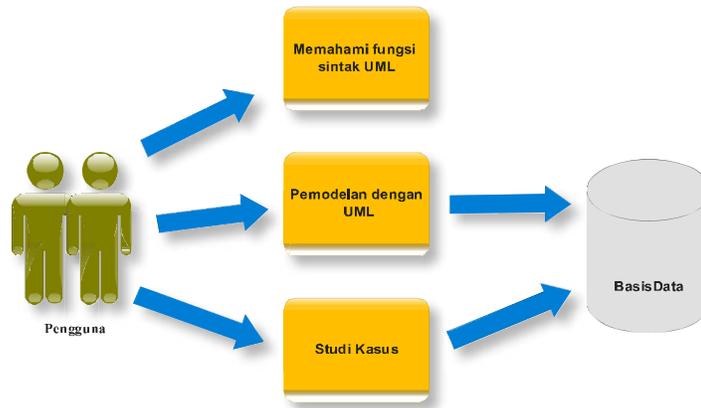
Gambar 1 menunjukkan tahapan dalam model prototyping. Tahapan awal dalam model ini adalah perencanaan yang dilakukan secara cepat kemudian dilanjutkan dengan perancangan. Setelah tahapan perancangan dilakukan, kemudian dilanjutkan dengan melakukan pembuatan prototype aplikasi. Setelah itu prototype akan diberikan kepada *customer* untuk direview dan diberikan umpan balik. Model *prototyping* memiliki keuntungan dalam hal komunikasi yang intens antara pengguna dan pengembang, membantu analisis dalam menentukan kebutuhan pengguna yang sebenarnya dan meminimalkan kesalahan persepsi.

14. ANALISIS DAN PERANCANGAN

Bagian ini membahas tentang analisis dan perancangan aplikasi pemodelan kebutuhan perangkat lunak.

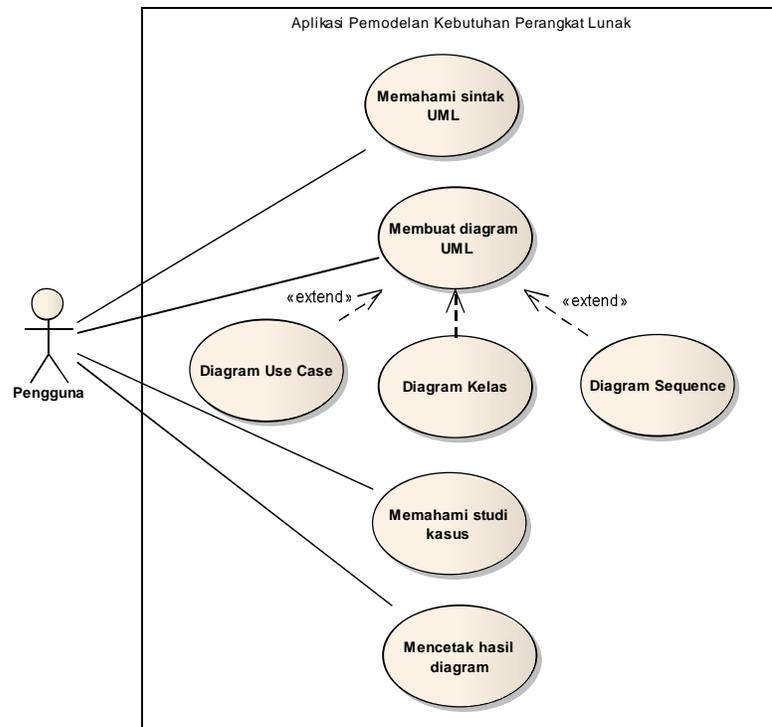
Analisis

Tahapan analisis dilakukan dalam rangka memodelkan masalah yang dijelaskan sebelumnya. Analisis yang dilakukan adalah analisis berorientasi objek menggunakan diagram-diagram dalam UML seperti diagram use case, diagram kelas, dan diagram aktivitas. Gambar 2 menunjukkan arsitektur aplikasi, yang terdiri dari 3 bagian utama yakni pengguna, menu utama dan basisdata. Menu utama aplikasi berisi penjelasan tentang sintak-sintak diagram dalam UML, pemodelan dengan diagram UML dan contoh studi kasus dan penyelesaiannya. Untuk pemodelan dengan UML dan studi kasus beserta penyelesaiannya akan tersimpan ke dalam basis data.



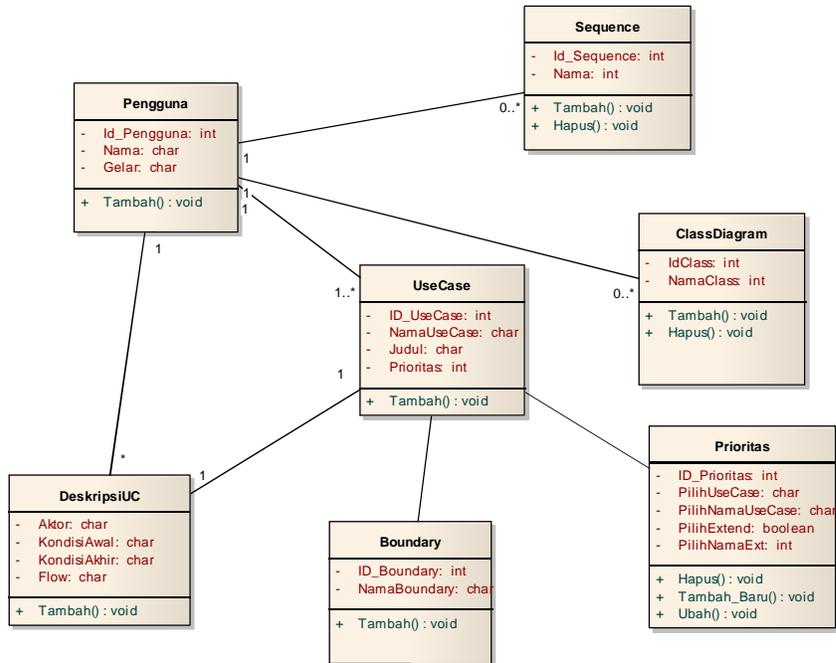
Gambar 2. Arsitektur aplikasi

Fungsionalitas aplikasi yang dibangun digambarkan dengan diagram use case seperti yang terlihat dalam gambar 3. Terdapat satu aktor yakni pengguna dan empat use case inti yakni memahami sintak UML, membuat diagram UML, memahami studi kasus dan penyelesaiannya, serta mencetak hasil diagram. Untuk membuat diagram UML dapat di *extend* menjadi beberapa diagram yang ada dalam UML, seperti diagram use case, diagram kelas, diagram sequence, dan diagram-diagram lainnya.



Gambar 3. Diagram use case aplikasi

Penggambaran jenis objek dan hubungannya dapat digambarkan dengan diagram kelas. Gambar 4 menunjukkan diagram kelas aplikasi yang akan dibangun, terdapat tujuh kelas utama yakni kelas pengguna, kelas use case, kelas sequence, kelas class, kelas prioritas, kelas boundary dan kelas deskripsi.



Gambar 4. Diagram kelas aplikasi

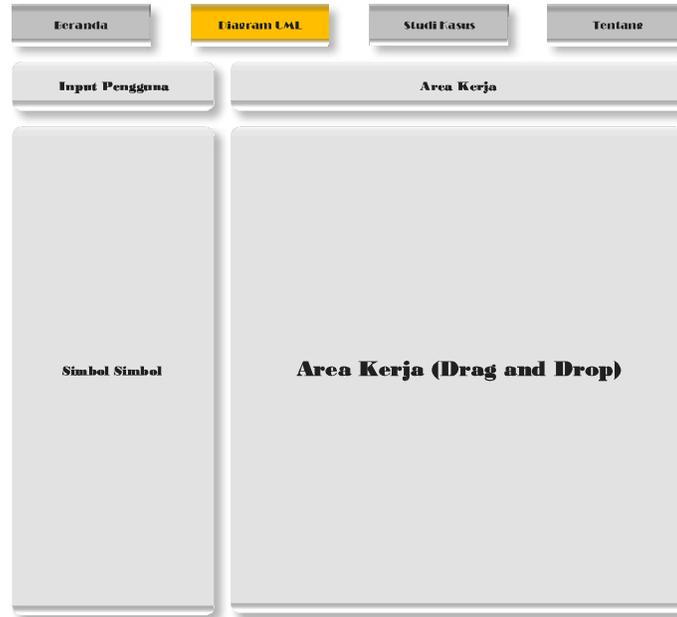
Perancangan

Bagian ini membahas tentang perancangan sistem yang akan dibangun. Perancangan lebih menekankan kepada perancangan antarmuka aplikasi. Gambar 5 menunjukkan perancangan halaman utama aplikasi yang dibangun. Halaman beranda menggambarkan deskripsi tentang diagram-diagram UML, lengkap dengan simbol-simbol dan penjelasan fungsi sintak-sintaknya.



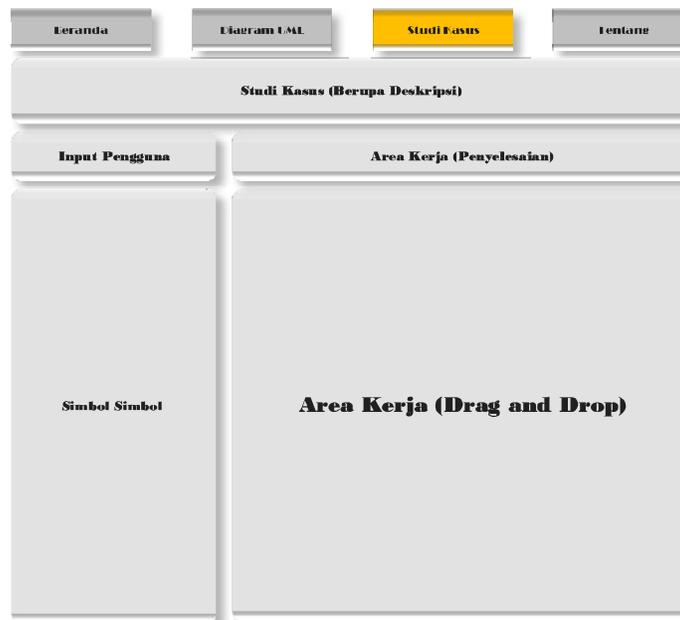
Gambar 5. Perancangan antarmuka halaman beranda

Gambar 6 menunjukkan perancangan antarmuka diagram UML, dalam menu ini terdapat drop down menu yang terdiri dari beberapa diagram, diantaranya adalah diagram use case, diagram kelas, diagram aktivitas, dan diagram sequence. Dalam menu ini disesuaikan dengan diagram yang dibangun, misalnya untuk pembuatan diagram kelas maka pada bagian input pengguna akan muncul simbol-simbol diagram yang akan dapat digunakan untuk di *drag* dan *drop* ke dalam area kerja, sehingga dihasilkan diagram yang diinginkan.



Gambar 6. Perancangan antarmuka halaman diagram UML

Gambar 7 menunjukkan perancangan antarmuka halaman studi kasus. Pada halaman ini ditunjukkan sebuah studi kasus, sehingga menuntun pengguna untuk dapat bekerja menggunakan aplikasi. Terdapat area untuk melakukan drag dan drop simbol-simbol disesuaikan dengan diagram yang dibangun.



Gambar 7. Perancangan antarmuka halaman studi kasus

15. KESIMPULAN

Pengembangan perangkat lunak meliputi beberapa tahapan pengembangan seperti perencanaan, analisis, perancangan, implementasi, pengujian, dan pemeliharaan. Khususnya dalam tahap analisis dilakukan pemodelan kebutuhan baik terstruktur maupun pemodelan berorientasi objek. UML telah menjadi standar de facto dalam pemodelan kebutuhan berorientasi objek. Kerumitan dan relatif mahal nya sebuah aplikasi menyebabkan kebingungan dalam diri seorang analis, khususnya analis pemula. Dalam dunia pendidikan, jarang sekali tersedia aplikasi khusus yang ditunjukkan untuk dunia pendidikan khususnya pengajaran. Dalam penelitian ini telah dianalisis dan dirancang sebuah aplikasi pemodelan kebutuhan perangkat lunak berorientasi objek. Penelitian berikutnya adalah melakukan implementasi pembangunan aplikasi sesuai dengan analisis dan perancangan yang telah dilakukan.

UCAPAN TERIMA KASIH

Penelitian ini dibiayai oleh 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.

PUSTAKA

- Alfert, K., & Schr, J. (2004). Software Engineering Education Needs Adequate Modeling Tools £ 1 Introduction 2 Lightweight Modeling Tools, 1–6.
- Auer, M., Tschurtschenthaler, T., & Biffi, S. (2003). A Flyweight UML Modelling Tool for Software Development in Heterogeneous Environments.
- Carrillo de Gea, J. M., Nicolás, J., Fernández Alemán, J. L., Toval, A., Ebert, C., & Vizcaíno, A. (2012). Requirements engineering tools: Capabilities , survey and assessment. *Information and Software Technology*, 54(10), 1142–1157. <http://doi.org/10.1016/j.infsof.2012.04.005>
- David, P., & Idasiak, V. (2009). Improving Reliability Studies With SysML.
- He, X. (2000). Formalizing UML Class Diagrams - A Hierarchical Predicate Transition Net Approach. *The 24th Annual International Computer Software and Applications Conference*. <http://doi.org/10.1109/CMPSAC.2000.884721>
- Khaled, L. (2009). A comparison between UML tools. In *2nd International Conference on Environmental and Computer Science, ICECS 2009* (pp. 111–114). <http://doi.org/10.1109/ICECS.2009.38>
- Mills, H. D. (1980). Software engineering education. *Proceedings of the IEEE*, 68(9), 1158–1162. <http://doi.org/10.1109/PROC.1980.11814>
- Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering A Practitioner's Approach 8th Edition*. <http://doi.org/10.1109/6.476732>
- Silingas, D. (2008). Towards Customizing UML Tools For Enterprise Architecture Modeling.
- Subhiyakto, E. R., Kamalrudin, M., Sidek, S., & S. Ahmad, S. S. (2014). Customization of Requirements Modelling Tools for Software Engineering Education. *International Symposium on Research in Innovation and Sustainability 2014*, 26(4), 1581–1584.
- Supakkul, S., & Chung, L. (2012). The RE-Tools: A multi-notational requirements modeling toolkit. In *2012 20th IEEE International Requirements Engineering Conference, RE 2012 - Proceedings* (pp. 333–334). Ieee. <http://doi.org/10.1109/RE.2012.6345831>