

APLIKASI PEMBELAJARAN CLASS DIAGRAM BERBASIS WEB UNTUK PENDIDIKAN REKAYASA PERANGKAT LUNAK

Egia Rosi Subhiyakto

Fakultas Ilmu Komputer, Program Studi Teknik Informatika
Universitas Dian Nuswantoro
Email: egia@dsn.dinus.ac.id

Yani Parti Astuti

Fakultas Ilmu Komputer, Program Studi Teknik Informatika
Universitas Dian Nuswantoro
Email: yani.parti.astuti@dsn.dinus.ac.id

ABSTRAK

Tahapan pembangunan perangkat lunak atau lebih dikenal dengan istilah SDLC (*Software Development Life Cycle*) memiliki beberapa tahapan termasuk tahapan analisis. Dalam tahap analisis pemodelan ke dalam bentuk diagram biasanya dilakukan secara mendalam untuk memodelkan masalah yang dihadapi. UML merupakan bahasa *de facto* dalam pemodelan kebutuhan perangkat lunak berorientasi objek. Aplikasi pemodelan yang berkembang mayoritas lebih ditujukan untuk dunia industri serta memerlukan instalasi dan lisensi yang terbatas. Penelitian ini mengembangkan sebuah aplikasi pemodelan kebutuhan perangkat lunak khususnya pemodelan *class* diagram. Aplikasi dikembangkan berbasis web, sehingga tidak perlu ada instalasi dan tidak perlu membayar lisensi. Fitur yang dikembangkan yakni pengenalan notasi-notasi *class* diagram beserta penjelasannya. Aplikasi juga menyediakan area kerja untuk menggambarkan diagram beserta fitur cetak hasil diagram yang telah dibuat. Aplikasi dikembangkan menggunakan metode pengembangan *prototyping*, sehingga didapatkan prototipe aplikasi. Prototipe aplikasi dapat terus dikembangkan sehingga mendapat hasil yang maksimal. Hasil pengujian fungsionalitas menggunakan metode pengujian *black box* didapatkan bahwa 100% fungsi berjalan dengan baik. Sedangkan pengujian *user acceptance* dengan tiga parameter didapatkan bahwa 92% responden setuju bahwa aplikasi memiliki kinerja yang baik, 90% responden setuju aplikasi mudah digunakan dan 89% responden puas terhadap aplikasi yang dibangun.

Kata kunci: UML; diagram; *class*; prototipe; aplikasi.

ABSTRACT

The stages of software development or better known as SDLC (Software Development Life Cycle) have several stages including the stages of analysis. In the analysis stage modeling in the form of diagrams is usually done in-depth to model the problem at hand. UML is a de facto language in modeling object-oriented software requirements. Application modeling that is developing today is more aimed at the industrial world and only a few parts that accommodate the world of education. This research develops a software requirements modeling application especially class diagram modeling. The application is developed based on the web, so there is no need for installation and no need to pay a license. The feature developed is the introduction of class diagram notations and their explanations. The application also provides a work area to illustrate the diagram along with the printed features of the diagram results that have been made. The application was developed using the prototyping development method so that an application prototype is obtained. The application prototype can be developed so that it gets maximum results. The results of testing the functionality using the black box testing method found that 100% of the functions run well. While user acceptance testing with three parameters found that 92% of respondents agreed that the application had good performance, 90% of respondents agreed that the application was easy to use and 89% of respondents were satisfied with the application built.

Keywords: UML; diagram; *class*; prototype; application.

1. PENDAHULUAN

Pengembangan perangkat lunak biasanya terbagi menjadi 2 bagian yakni tujuan industri dan tujuan pendidikan. Perangkat lunak yang ditujukan untuk dunia industri tergolong kompleks dan biasanya berbayar. Sedangkan untuk dunia pendidikan, pengembangan perangkat lunak relatif lebih sederhana agar mudah dipahami oleh mahasiswa. Mahasiswa yang mempelajari dunia rekayasa perangkat lunak biasanya tergolong sebagai pengembang perangkat lunak pemula atau *novice developer*. Pengetahuan mengenai pengembangan perangkat lunak wajib dimiliki oleh mahasiswa agar menjadi pengembang perangkat lunak yang memiliki kompetensi [1].

Tahapan pengembangan perangkat lunak meliputi komunikasi, perencanaan, pemodelan, konstruksi dan *deployment* [2]. Tahap pemodelan merupakan bagian inti dalam memodelkan masalah maupun solusi perangkat lunak yang akan di bangun.

UML telah menjadi standar *de facto* bahasa pemodelan perangkat lunak berorientasi objek. Aplikasi pemodelan yang digunakan banyak jenisnya, perbandingan aplikasi pemodelan UML telah dilakukan oleh [3] yang membandingkan aplikasi seperti Rational Rose, ArgoUML, MagicDraw, dan Enterprise Architect. Sedangkan dalam [4] juga membandingkan aplikasi seperti Rational Rose, Enterprise Architect, MS. Visio, Magic Draw, UMLet, STORM, dan RMTTool. Dari hasil perbandingan didapatkan bahwa mayoritas aplikasi ditujukan untuk dunia industri dan berbayar. Pengembangan aplikasi pemodelan kebutuhan perangkat lunak telah dilakukan oleh beberapa peneliti, diantaranya adalah [5] yang mengembangkan sebuah alat pemodelan UML berbasis web dengan menggunakan metode SCRUM, alat yang dikembangkan tidak membutuhkan instalasi dan spesifikasi yang cukup ringan agar dapat menggunakan aplikasi. Penelitian lain dalam [6] mengembangkan sebuah LogViz untuk memantau aktifitas mahasiswa dalam merancang UML.

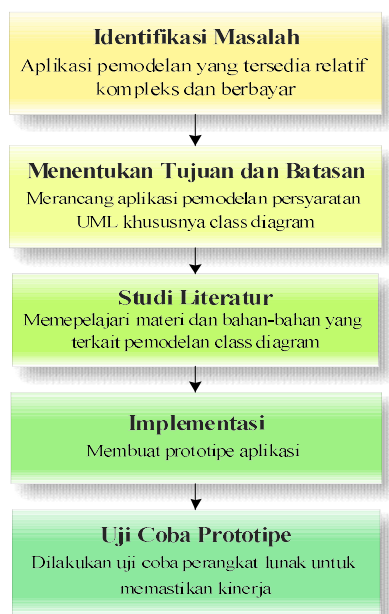
Penelitian yang dilakukan merupakan hasil pengembangan dari penelitian terdahulu yang memiliki keterbatasan dalam hal diagram yang dimodelkan. Jika sebelumnya hanya use case diagram pada penelitian kali ini adalah mengembangkan aplikasi pemodelan *class diagram* berbasis web sebagai solusi dari kompleksitas dan lisensi yang berbayar dari aplikasi pemodelan yang telah ada. Aplikasi dikembangkan khusus untuk memodelkan diagram kelas yang ada dalam UML. Aplikasi memiliki fitur pengenalan notasi-notasi *class diagram* beserta penjelasannya. Aplikasi juga menyediakan area kerja untuk menggambarkan diagram beserta fitur cetak hasil diagram yang telah dibuat. Aplikasi dikembangkan dalam platform web, sehingga tidak perlu proses instalasi dan mudah diakses.

2. METODOLOGI PENELITIAN

Tahapan penelitian mencakup lima tahapan seperti yang terdapat dalam gambar 1. Tahap penelitian dapat dijelaskan sebagai berikut:

Masalah yang dihadapi adalah aplikasi pemodelan yang tersedia di pasaran relatif kompleks dan berbayar. Hal tersebut menjadi alasan utama kami mengembangkan aplikasi pemodelan yang gratis dan mudah dipahami. Aplikasi yang dibangun nantinya berbasis web dan tidak perlu instalasi. Tujuan penelitian ini adalah mengembangkan aplikasi pemodelan kebutuhan perangkat lunak khususnya *class diagram*. Aplikasi berbasis web dibangun menggunakan bahasa pemrograman PHP, HTML dan CSS. Terdapat fitur penjelasan notasi *class diagram* dan terdapat area kerja untuk membuat diagram serta fitur pencetakan diagram. Untuk studi literatur digunakan untuk mencari materi terkait dengan pemodelan kebutuhan perangkat lunak khususnya *class diagram*. Terdapat beberapa jurnal, buku dan dokumen prosiding yang digunakan sebagai literatur.

Fase implementasi dilakukan dengan menggunakan perangkat lunak pendukung seperti XAMPP, Sublime text, dan notepad++. Sedangkan untuk bahasa pemrograman menggunakan PHP, HTML, dan CSS. Aplikasi dibuat menjadi prototipe yang dapat di evaluasi oleh pengguna secara langsung. Uji coba prototipe dilakukan oleh pengguna secara langsung, apabila terdapat kekurangan berdasarkan hasil evaluasi maka akan dilakukan perbaikan. Evaluasi prototipe untuk memastikan kinerja aplikasi, kemudahan penggunaan dan tingkat kompleksitas aplikasi.



Gambar 1. Tahapan Penelitian

3. HASIL DAN PEMBAHASAN

Bagian hasil dan pembahasan meliputi analisis, perancangan, implementasi dan pengujian serta evaluasi aplikasi yang dibangun.

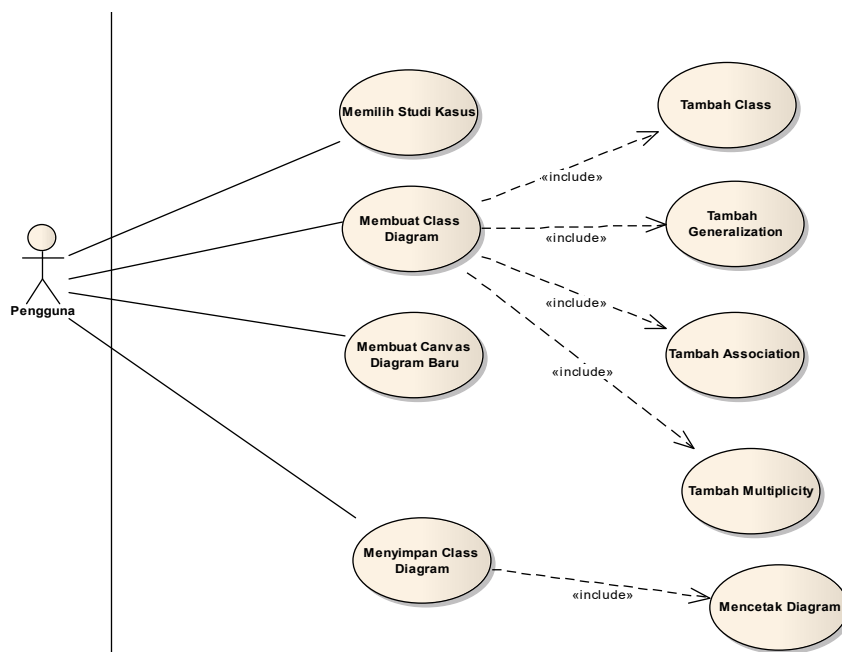
3.1 Analisis

Analisis dilakukan untuk mengetahui kebutuhan-kebutuhan apa yang diperlukan dalam sebuah aplikasi, dalam tabel 1 ditunjukkan kebutuhan fungsional aplikasi pemodelan kebutuhan.

Tabel 1. Kebutuhan fungsional

Kebutuhan	Deskripsi
Fungsional 1	Sistem dapat menambahkan element class
Fungsional 2	Sistem dapat menambahkan element generalitation
Fungsional 3	Sistem dapat menambahkan element assosiation
Fungsional 4	Sistem dapat menambahkan element multiplicity
Fungsional 5	Sistem dapat menyimpan diagram dalam format jpg atau png
Fungsional 6	Sistem dapat menampilkan studi kasus yang disediakan

Gambar 2 menunjukkan use case diagram aplikasi, terdapat satu orang aktor yakni pengguna. Sedangkan *use case* sistem terdiri dari *use case* memilih studi kasus, *use case* membuat *class* diagram, *use case* membuat diagram baru, dan *use case* menyimpan hasil diagram.



Gambar 2. Use Case Diagram Aplikasi

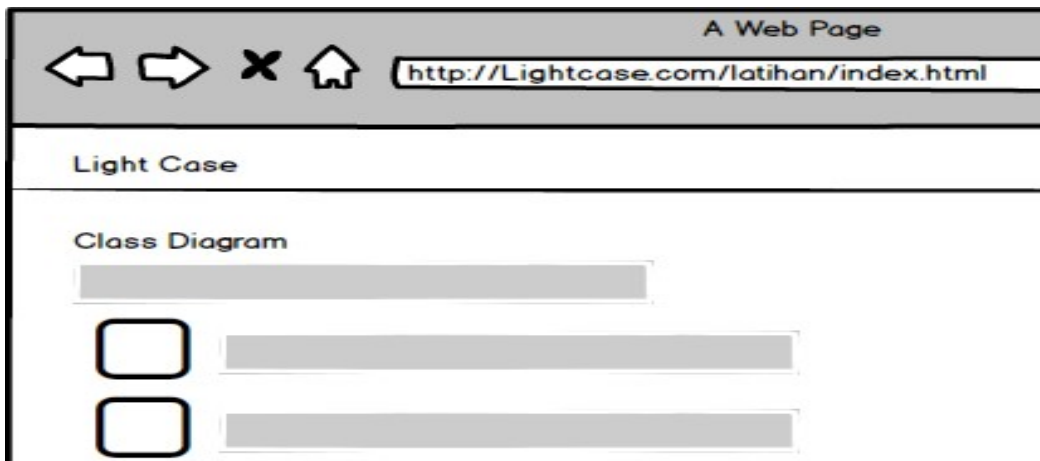
3.2 Perancangan

Perancangan dilakukan dengan melakukan perancangan antarmuka aplikasi. Gambar 3 menunjukkan halaman utama dari aplikasi yang terdiri dari tentang aplikasi dan bagaimana menggunakannya. Terdapat beberapa objek menu diantaranya adalah tentang aplikasi, tutorial aplikasi, dan demo aplikasi.

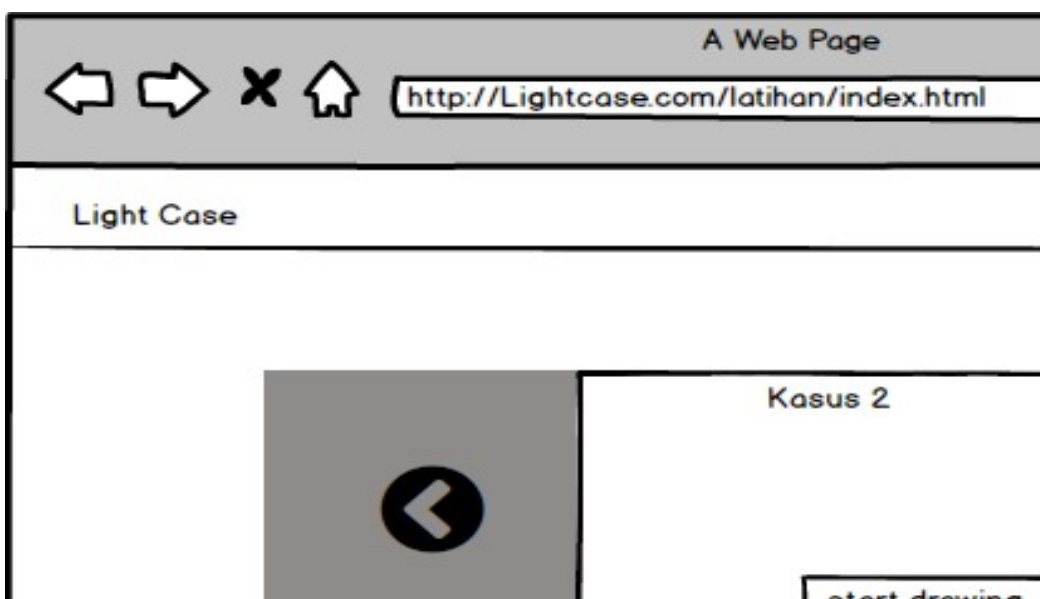


Gambar 3. Perancangan Halaman Utama

Gambar 4 menunjukkan perancangan halaman notasi, dalam halaman ini terdapat bagian notasi *class* diagram dan penjelasannya. Selain itu juga terdapat link menuju tentang aplikasi, tutorial dan demo.



Gambar 4. Perancangan Halaman Notasi

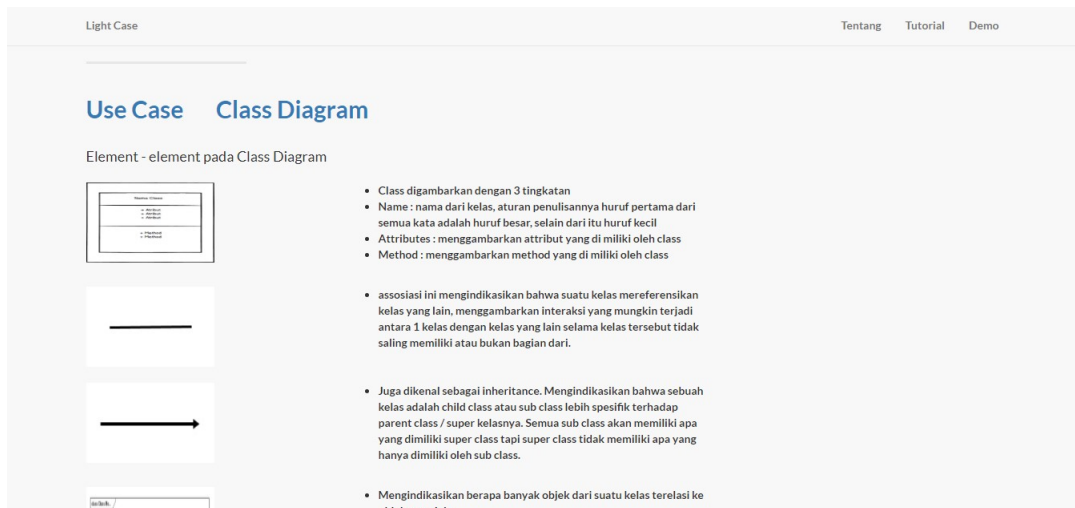


Gambar 5. Perancangan Halaman Pembuatan Diagram

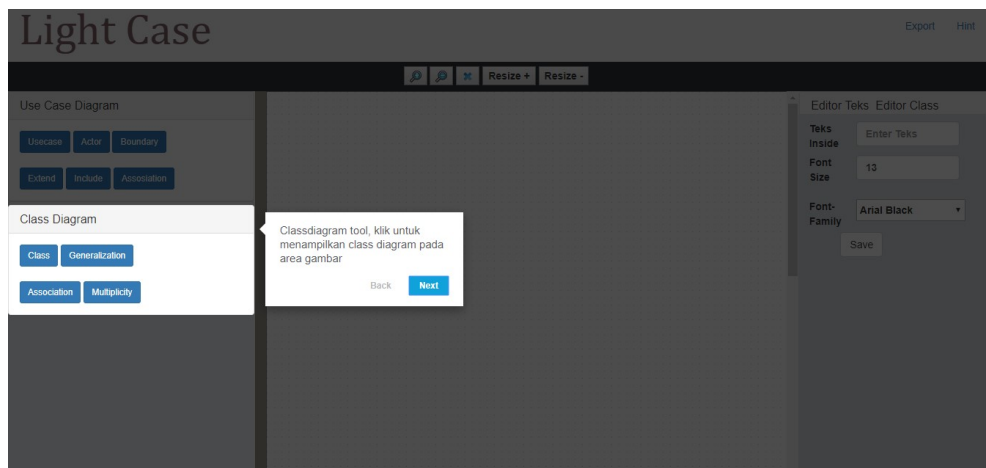
Gambar 5 menunjukkan perancangan halaman pembuatan diagram, terdapat tombol klik *start drawing* untuk memulai membuat diagram, kemudian pengguna dapat membuat *class diagram* sesuai notasi.

3.3 Implementasi

Implementasi adalah melakukan pengkodean aplikasi, gambar 6 merupakan hasil implementasi halaman utama. Terdapat elemen notasi-notasi terkait *class diagram* beserta penjelasannya.

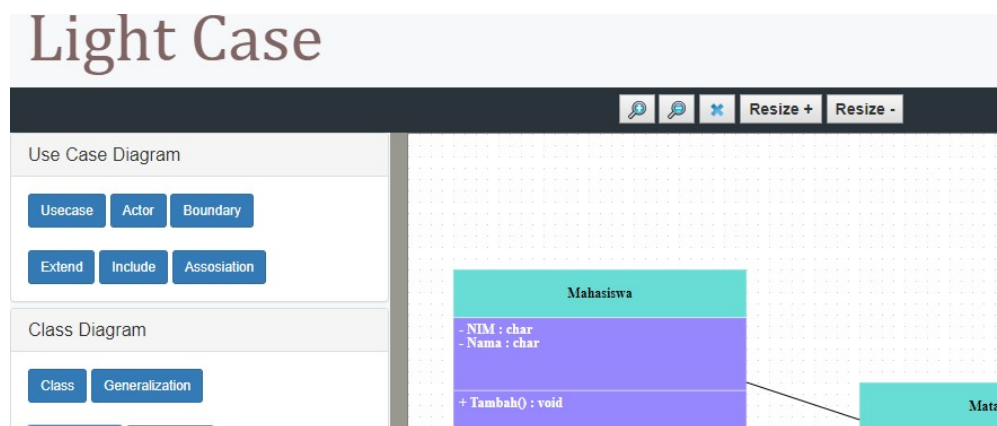


Gambar 6. Implementasi Halaman Utama



Gambar 7. Implementasi Halaman Area Kerja

Gambar 7 menunjukkan implementasi halaman area kerja, terdapat beberapa notasi seperti notasi *class*, notasi asosiasi, notasi generalisasi dan notasi multiplicity. Bagian tengah merupakan area kerja di mana dapat digunakan untuk membuat diagram. Sedangkan di bagian kanan terdapat editor teks untuk mengedit nama *class*, nama atribut, nama method, mengubah ukuran font dan mengubah jenis font.



Gambar 8. Implementasi Halaman Pembuatan Diagram

Gambar 8 merupakan implementasi halaman pembuatan diagram, dalam gambar tersebut di contohkan cara pembuatan diagram yang mana sebelah kiri adalah notasi, tengah adalah area kerja pembuatan diagram dan kanan adalah editor teks.

3.4 Pengujian

Pengujian fungsionalitas aplikasi menggunakan pengujian *black box*. Teknik dan strategi pengujian terdapat dalam [7]. Terdapat beberapa fungsi yang diuji yang ditunjukkan dalam tabel 2.

Tabel 2. Identifikasi dan rencana pengujian

<i>No</i>	<i>Nama Fungsi</i>	<i>Jenis Pengujian</i>
1.	Tentang	<i>Black Box</i>
2.	Tutorial	<i>Black Box</i>
3.	Class	<i>Black Box</i>
4.	Demo	<i>Black Box</i>
5.	Pelajari	<i>Black Box</i>
6.	Export	<i>Black Box</i>
7.	Hint	<i>Black Box</i>
8.	Class Diagram	<i>Black Box</i>
9.	Tambah Class	<i>Black Box</i>
10.	Tambah Generalitation	<i>Black Box</i>
11.	Tambah Association	<i>Black Box</i>
12.	Tambah Multiplicity	<i>Black Box</i>
13.	Editor Text	<i>Black Box</i>
14.	Editor Class	<i>Black Box</i>
15.	Tambah nama class	<i>Black Box</i>
16.	Tambah Atribut	<i>Black Box</i>
17.	Tambah Method	<i>Black Box</i>

Pengujian blackbox kemudian dilakukan sesuai dengan identifikasi, tabel 3 menunjukkan prosedur pengujian dan hasil atau kesimpulan pengujian secara *black box*.

Tabel 3. Prosedur dan hasil pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran	Kesimpulan
Tentang	Menampilkan halaman tentang.	Mengklik button tentang	Klik	Halaman tentang	Diterima
Tutorial	Menampilkan tutorial tentang	Mengklik button tutorial	Klik	Halaman tutorial	Diterima
Class	Menampilkan halaman tutorial	Mengklik button class	Klik	Halaman tutorial	Diterima

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran	Kesimpulan
Demo	Menampilkan halaman kasus dan	Klik button demo	Klik	Halaman kasus	Diterima
Pelajari	Menampilkan halaman cara	Klik button pelajari	Klik	Halaman penggunaa	Diterima
Export	Menampilkan halaman diagram	Klik button export	Klik	Halaman diagram	Diterima
Hint	Menampilkan bantuan dalam	Klik button hint	Klik	Pop up bantuan	Diterima
Class Diagram	Menampilkan menu class	Klik button class diagram	Klik	Menu class diagram	Diterima
Tambah Class	Menampilkan dan menambah class	Klik button class	<i>Drag and Drop</i>	Tampilan class	Diterima
Tambah Generalitati	Menampilkan generalitation	Klik button generalitation	<i>Drag and Drop</i>	Tampilan generalitati	Diterima
Tambah Association	Menambah association	Klik button association	<i>Drag and Drop</i>	Tampilan association	Diterima
Tambah Multiplicity	Menambah multiplycity	Klik button multiplycity	<i>Drag and Drop</i>	Tampilan Multiplycit	Diterima
Editor Text	Menampilkan menu untuk edit usecase	Klik button editor text	<i>Edit Text</i>	Tampilan menu edit	Diterima
Editor Class	Menampilkan menu untuk class	Klik button editor class	<i>Edit Text</i>	Tampilan menu edit	Diterima
Edit nama class	Mengedit nama class	Ketik nama class	<i>Edit Text</i>	Tampilan edit nama	Diterima
Tambah Atribut	Menambahkan atribut	Ketik nama atribut baru	<i>Drag and Drop</i>	Tampilan tambah	Diterima
Tambah Method	Menambahkan method	Klik Method Ketik nama	<i>Drag and Drop</i>	Tampilan Tambah	Diterima

Berdasarkan hasil pengujian dari tabel 3 di atas didapatkan bahwa semua fungsi sudah lolos uji dengan pengujian *black box*, sehingga dapat disimpulkan bahwa 100% fungsi sudah sesuai harapan tanpa adanya *error*. Selanjutnya pengujian dilakukan kepada pengguna aplikasi dengan memberikan kuisioner kepada 30 orang responden. Responden diberikan 10 pertanyaan dengan skala Likert 1-5. Skala likert mencakup parameter sangat setuju (SS), setuju (S), netral (N), tidak setuju (TS), dan sangat tidak setuju (STS). Prosentase kemudian akan didapatkan berdasarkan jumlah perhitungan poin rata-rata dibagi dengan jumlah parameter.

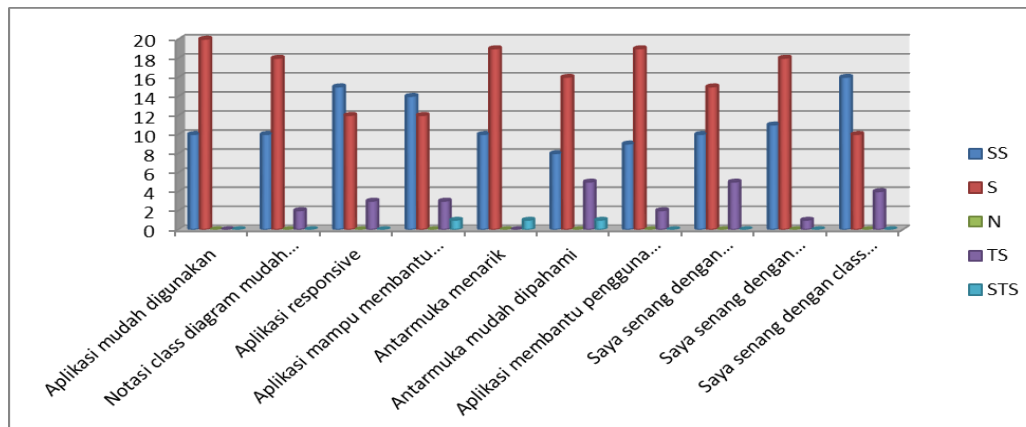
Tabel 4. Kuisisioner

No	Pernyataan	SS	S	N	TS	STS
1	Aplikasi mudah digunakan	10	20	0	1	0
2	Notasi class diagram mudah dipahami	10	18	0	3	0
3	Aplikasi responsive	15	12	0	4	0
4	Aplikasi mampu membantu mengenalkan class diagram	13	12	0	4	1
5	Antarmuka menarik	10	19	0	1	1
6	Antarmuka mudah dipahami	8	16	0	6	1
7	Aplikasi membantu pengguna mengubah dan memodifikasi	9	19	0	3	0
8	Saya senang dengan antarmuka aplikasi	10	15	0	6	0
9	Saya senang dengan fungsionalitas aplikasi	11	18	0	2	0
10	Saya senang dengan class diagram yang dibuat	16	10	0	5	0

Hasil kuisioner ditunjukkan dalam tabel 4, terdapat 10 pernyataan dengan tiga parameter yakni pernyataan 1-4 adalah pernyataan kinerja aplikasi, pernyataan 5-7 adalah pernyataan kinerja kemudahan penggunaan, dan pernyataan 8-10 adalah pernyataan kepuasan pengguna.

Gambar 9 menunjukkan grafik hasil pengujian penerimaan pengguna, berdasarkan hasil kuisioner didapatkan bahwa 92% responden sangat setuju dan setuju bahwa aplikasi memiliki kinerja yang baik,

90% sangat setuju dan setuju bahwa system mudah digunakan dan 89% responden puas dengan aplikasi yang dibangun.



Gambar 9. Grafik Hasil Pengujian *User Acceptance*

4. KESIMPULAN

Class diagram merupakan salah satu diagram yang terdapat dalam UML. Dalam tahap analisis terdapat bagian untuk memodelkan masalah dengan visualisasi berupa diagram. Banyaknya aplikasi yang beredar dewasa ini memiliki kompleksitas serta kekurangan dan kelebihan masing-masing. Dalam penelitian ini telah dilakukan perancangan dan pembangunan prototipe aplikasi untuk memodelkan *class diagram*. Aplikasi yang dibangun berbasis web dan tidak memerlukan instalasi. Berdasarkan hasil pengujian didapatkan bahwa fungsionalitas aplikasi telah berjalan dan tidak ditemukan *error*. Sedangkan berdasarkan hasil evaluasi pengguna didapatkan bahwa aplikasi yang dibangun mayoritas pengguna puas dengan kinerja aplikasi, kemudahan penggunaan dan puas terhadap keseluruhan fungsionalitas dan antarmuka aplikasi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih pada LPPM Udinus Semarang yang telah memberikan dukungan finansial terhadap penelitian ini, terimakasih juga diucapkan untuk editor, reviewer, dan semua pihak yang telah membantu dalam penyelesaian artikel ini.

DAFTAR PUSTAKA

- [1] E. R. Subhiyakto, 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] R. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. Yogyakarta, 2015.
- [3] L. Khaled, "A Comparison between UML Tools," *2009 Second Int. Conf. Environ. Comput. Sci.*, pp. 111–114, 2009.
- [4] E. R. Subhiyakto and D. W. Utomo, "RMTTool; Sebuah Aplikasi Pemodelan Persyaratan Perangkat Lunak menggunakan UML," *JNTETI*, vol. 6, no. 3, pp. 268–274, 2017.
- [5] M. Subekti, D. Indrawan, and G. Putra, "Perancangan Case Tools untuk Diagram Use Case, Activity, dan *Class* untuk Permodelan UML Berbasis Web Menggunakan HTML5 dan PHP," *ComTech*, vol. 5, no. 2, pp. 625–635, 2014.
- [6] D. R. Stikkolorum, T. Ho-quang, and Chaudron M.R.V, "Revealing Students' UML *Class* Diagram Modelling Strategies with WebUML and LogViz," in *41st Euromicro Conference on Software Engineering and Advanced Applications Revealing*, 2015.
- [7] E. R. Subhiyakto and D. W. Utomo, "Software Testing Techniques and Strategies Use in Novice Software Teams," *J. Sist. Inf.*, 2016.