

PERANCANGAN CASE TOOLS UNTUK DIAGRAM USE CASE, ACTIVITY, DAN CLASS UNTUK PERMODELAN UML BERBASIS WEB MENGGUNAKAN HTML5 DAN PHP

Mohammad Subekti; Lukman; Donny Indrawan; Ganesh Putra

Computer Science Department, School of Computer Science, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
subekti@binus.ac.id

ABSTRACT

This study is intended to generate an application tools (CASE tools) that allows a software developer to create a modeling system design using Unified Modeling Language (UML), especially in making use case, activity or class diagrams more quickly and easily. The tools developed will also facilitate developers in doing UML modeling by accessing the network through a web-based internet application. With the web-based applications, the users require only a browser and an internet connection to use this application. This application also helps developers to understand of how to make UML diagrams correctly and good. In this research traditional methods Scrum model is used. Scrum method is Agile methods that is a process to cultivate software easily and can be developed in accordance with the development of information technology. Scrum is using empirical methods or in other words every stage in it involves inspection and adaptation.

Keywords: scrum method, agile method, case tools, UML diagram, use case diagram, activitydiagram, classdiagram

ABSTRAK

Penelitian ini adalah dimaksudkan untuk membuat sebuah aplikasi alat bantu (CASE tool) yang memungkinkan seorang pengembang dapat membuat rancangan sistem dengan permodelan Unified Modeling Language (UML) khususnya dalam membuat diagram-diagram use case, activity atau class dengan lebih cepat dan mudah. Alat bantu yang dikembangkan, juga dapat mempermudah pengembang dalam melakukan pengaksesan dan pembuatan model UML melalui jaringan internet karena dibuat berbasis web. Dengan berbasis web, maka untuk menjalankan aplikasi ini pengguna hanya memerlukan perangkat lunak browser dan koneksi internet. Aplikasi ini juga membantu pengembang untuk memahami pembuatan diagram UML yang benar dan baik. Penelitian yang dilakukan tim penulis menggunakan metode Scrum Model. Metode Scrum, adalah metode Agile yang merupakan proses untuk mengolah perangkat lunak dengan mudah serta dapat dikembangkan sesuai dengan perkembangan teknologi informasi. Scrum menggunakan metode empiris atau dengan kata lain setiap tahap di dalamnya melibatkan inspeksi dan adaptasi.

Kata kunci: metoda scrum, metoda agile, CASE tools, UML, diagram Use Case, diagram activity, diagram class

PENDAHULUAN

Saat ini berbagai piranti lunak semakin luas penggunaannya, baik untuk sistem yang sederhana maupun untuk sistem yang kompleks. Piranti lunak diharapkan menghasilkan luaran yang baik, sehingga perlu dibuat dengan menggunakan pemodelan (*modeling*) yang sebaik mungkin dan memerlukan penanganan yang jelas jika terjadi *error* atau *bug*.

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena untuk memahami sistem yang kompleks secara menyeluruh sangat sulit. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebutan segitiga sukses (*the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan *tool* yang digunakan.

CASE tools merupakan teknik yang digunakan untuk membantu melakukan beberapa fase, yakni fase analisis, desain, *maintenance*, dan implementasi dalam perancangan piranti lunak. Dengan menggunakan CASE tools maka dapat memperbesar kemungkinan otomatisasi pada setiap *phase life-cycle* piranti lunak dan sangat membantu dalam meningkatkan kualitas rancangan model suatu piranti lunak sebelum piranti lunak itu dibangun/dikembangkan, baik piranti lunak yang dibangun dalam *environment* yang sederhana maupun kompleks. Kebanyakan CASE tools yang digunakan berbasis *desktop* sehingga memerlukan instalasi yang cukup panjang pada setiap komputer yang ingin digunakan, terbatas pada sistem operasi tertentu, media yang terbatas untuk mengakses aplikasi tersebut, dan memerlukan spesifikasi komputer yang tinggi untuk menjalankan aplikasi.

Melihat kelemahan-kelemahan pada aplikasi CASE tools berbasis *desktop*, maka penulis merancang aplikasi CASE tools berbasis *Web*. Dengan CASE tools berbasis *Web*, maka masalah-masalah tersebut di atas dapat diselesaikan dan perancangan piranti lunak dapat dibuat secara lebih efisien. Aplikasi berbasis *Web* dapat dijalankan tanpa instalasi piranti lunak khusus untuk menggunakannya, melainkan hanya memerlukan *browser* dan akses internet. CASE tools berbasis *Web* dapat digunakan di mana saja dan kapan saja, dijalankan pada sistem operasi apa saja, dan diakses melalui banyak media seperti *handphone* dan komputer yang sudah sesuai dengan standar WAP, serta tidak memerlukan spesifikasi tinggi untuk menggunakan aplikasi berbasis *Web* ini karena sebagian besar proses dilakukan di *website server* penyedia aplikasi.

Melihat kebutuhan dan teknologi yang ada, maka dibutuhkan aplikasi CASE tools untuk pemodelan UML yang handal dan mempunyai fungsionalitas standar UML dengan menggunakan HTML5 yang memiliki integrasi yang lebih baik. Adapun fitur yang akan dibuat adalah *Save* dan *Open Project* secara *online* pada *website*, dan *Multiple Diagram in one Project*.

Penelitian dimaksudkan untuk menjawab berbagai permasalahan dalam proses pengembangan aplikasi sistem informasi modern, antara lain: Aplikasi berbasis *web* seperti apa yang dapat mempercepat pembuatan pemodelan UML secara efektif dan efisien? Bagaimana aplikasi berbasis *web* ini dapat menyimpan hasil pemodelan UML lebih baik dibanding aplikasi *desktop*? Bagaimana aplikasi berbasis *web* ini dapat menghubungkan satu diagram dengan diagram lain pada saat dijalankan di *browser*? Bagaimana aplikasi berbasis *Web* ini dapat dijalankan lebih ringan dari pada aplikasi *desktop*?

Dari berbagai pertanyaan tersebut dapat diformulasikan kemungkinan sebuah rancangan alat bantu untuk pengembangan perangkat lunak yang mempunyai karakteristik-karakteristik sebagai berikut, yaitu: sebuah aplikasi alat bantu pembuatan model UML berbasis *Web*, yang lebih efisien dan efektif dibandingkan dengan aplikasi *desktop* yang sekarang ini ada. Alat bantu harus memiliki fitur

Save dan *Open Project* secara *online* di jaringan internet yang memungkinkan pengembang menyimpan hasil pembuatan pemodelan UML tersebut berdasarkan lokasi objek diagram yang dibuat, sehingga pada saat dibuka kembali *project* tersebut dapat ditampilkan kembali pada *web browser* seperti yang telah penulis buat sebelumnya. Alta bantu juga harus memiliki fitur *Drag and Drop* di dalam *browser* untuk memudahkan pembuatan UML hanya dengan melakukan *drag and drop* seperti halnya pada *desktop*. Yang terakhir, alat ini harus memiliki fitur *Multiple Diagram in one Project* memungkinkan *user* membuat banyak jenis diagram UML dalam satu *project*.

Penelitian ini ditujukan untuk menghasilkan CASE tool UML yang dijalankan secara *online* yang memungkinkan untuk bekerja lebih cepat dan efisien. Bagi *user*, dapat mempermudah pembuatan pemodelan UML karena hanya memerlukan *browser* dan koneksi internet maka CASE tool dapat langsung digunakan. Perancangan aplikasi ini dibatasi pada pembuatan model data konseptual dan model data fisik dari UML, antara lain: *class diagram*, *use case diagram*, dan *activity diagram*. Aplikasi ini menyediakan fasilitas *online save* karena berbasis *website* sehingga *user* dapat mengakses UML nya di mana saja dan *user* lain pun dapat mengaksesnya.

METODE

Metode penelitian yang digunakan adalah permodelan Scrum (Sutherland, 2010) yang merupakan salah satu jenis Agile Piranti lunak *Development Method*. Menurut Cervone (2011), Scrum adalah Agile itu sendiri yang merupakan proses yang ringan untuk mengelola dan mengendalikan pengembangan piranti lunak dan produk dalam lingkungan yang berubah dengan cepat. Pada permodelan Scrum ada *roles/peran*, *artifacts* dan *process* yang penting untuk dipahami.

Scrum Roles terdiri dari: *Scrum Master*, peran dari *Scrum Master* adalah membantu *Scrum Team* supaya memberikan performa terbaik dan mengatasi hambatan yang dihadapi dalam proses pengerjaan. *Scrum Master* tidak menentukan tugas *Scrum Team*, melainkan fokus kepada *team* dan tantangan yang dihadapi. *Scrum Team* merupakan tim yang bekerja secara mandiri dan menentukan tugas-tugas yang akan diselesaikan di dalam *sprint*. *Product Owner* adalah *stakeholders* yang tahu apa yang harus dibangun dan bagaimana seharusnya *progress* berjalan.

Sedangkan *Scrum Artifacts* adalah terdiri atas: *Product backlog* merupakan daftar kebutuhan *backlog items*, yaitu fitur dan produk yang akan diselesaikan dan diatur berdasarkan skala prioritas. *Sprint backlog* merupakan daftar tahap pengerjaan untuk penyelesaian *backlog items* yang dibuat secara mandiri oleh *Scrum Team*. *Burn down charts* adalah diagram yang digunakan untuk mengindikasikan *progress* dari keseluruhan *project*.

Berikut adalah proses kerja dalam *Scrum Process*: (1) *Kickoff Meeting* adalah pertemuan yang dihadiri oleh *Scrum Master*, *Scrum Team*, dan *Product Owner* untuk menentukan tujuan utama proyek. (2) *Sprint Planning* adalah *meeting* yang dihadiri oleh *Scrum Master*, *Scrum Team*, dan *Product Owner* diawal setiap iterasi(*sprint*). Fokus *meeting* ini ada dua hal yaitu: pertama menentukan *product backlog* dan hasil akhir *sprint*. Kedua, menentukan *sprint backlog*. (3) *Sprint* merupakan proses pengerjaan *product backlog* sesuai dengan *sprint backlog* yang dilakukan oleh *Scrum Team*. Dalam proses pengerjaan akan berjalan secara tertutup dan tidak boleh ada perubahan *requirements*. (3) *Daily Scrum* adalah pertemuan singkat sepenulis 15 menit yang dipimpin oleh *Scrum Master* sebelum memulai *sprint* setiap harinya. Pertanyaan yang biasa ditanyakan pada pertemuan ini antar: Apa yang sudah dilakukan pada Scrum terakhir? Apa yang akan kalian lakukan untuk Scrum berikutnya? Apa yang membuat pekerjaan kalian tertunda? Tujuan dari pertemuan singkat ini adalah melacak *progress* dari *sprint* dan membentuk komitmen yang kuat antar anggota tim. (4) *Sprint Review Meeting* dilaksanakan pada akhir *sprint* dan mendemonstrasikan hasil *sprint* kepada *Product Owner*.

Tinjauan Pustaka

Pada umumnya CASE tools dijalankan dengan berbasis *desktop*, dan tidak dapat diakses di *website*. CASE tool berbasis Web merupakan alat yang dapat membantu mengatasi masalah ini karena memberikan pengembangan yang didistribusikan secara geografis sehingga pengembang dapat berkolaborasi dan bekerja sama dalam proyek piranti lunak secara lebih efisien. (Palaniappan, Ling, 2008).

Banyak tools UML yang sudah ada saat ini terlalu berat untuk menghasilkan model yang sederhana dengan cepat. Tools UML seperti ini biasanya mengandung banyak fungsionalitas, yang dapat menyebabkan masalah kegunaan dan produktivitas yang kurang. Tools UML yang ringan dirancang untuk mendukung pembuatan model yang cepat dan mudah. Penggunaan *website* telah meningkatkan manfaat dalam pembuatan model termasuk portabilitas dan kolaborasi penggunaan. (Mackay, Noble, Biddle, 2002).

Pendekatan tradisional untuk menyediakan tools diagram seperti CASE tools, CAD tools dan tools desain antarmuka adalah program yang menggunakan teknologi *thick client*, di mana program dapat dijalankan pada komputer *host* dan menggunakan fasilitas *desktop* grafis. Keunggulan utamanya adalah interaksi yang sangat responsif ketika langsung memanipulasi diagram dan kemampuan alat pengembang untuk menggunakan semua fasilitas grafis yang tersedia pada komputer *host*. Namun, *tools diagram* yang masih tradisional memiliki masalah umum, antara lain aplikasi perlu di-*install* dan secara berkala diperbaharui di semua komputer *host*, *user interface* yang kompleks, dan membutuhkan infrastruktur yang cukup kompleks untuk mendukung *multi-user* kolaboratif, terutama sinkronisasi pembuatan diagram. (Cao *et al*, 2005).

Dengan melihat berbagai pendekatan dari jurnal serta permasalahan yang ada, maka perlu dibuat sebuah aplikasi CASE tools berbasis Web dengan menggunakan HTML 5 sebagai aplikasi yang handal, ringan, serta mudah digunakan dan memiliki *user interface* yang *friendly* untuk dapat memudahkan *user* dan *system analyst* yang hendak membuat pemodelan UML dengan menggunakan CASE tools yang memiliki banyak kelebihan.

HASIL DAN PEMBAHASAN

Analisis Aplikasi

Untuk analisis aplikasi ini, penulis mengumpulkan informasi dengan membandingkan jurnal-jurnal penelitian ilmiah yang berkaitan dengan topik Case Tools berbasis Web, dalam sub-bab ini akan dijelaskan tentang permasalahan yang dihadapi peneliti sebelumnya terkait dengan perancangan aplikasi sejenis. Penulis mengambil tiga penelitian yang paling mendekati dengan topik perancangan Case Tools berbasis web sebagai berikut: (1) Alat bantu pengembangan perangkat lunak CASE Berbasis Web untuk *Automated Rendering* UML. (2) Menggunakan antar muka pengguna berbasis Web User untuk penggambaran *Diagram Model*. (3) Alat bantu pengembangan ringan (*light weight*) untuk diagram UML dijalankan di semua jenis platform.

Pada penelitian sebelumnya, Palaniappan (2008) memaparkan tujuan dari perancangan Case Tools berbasis Web adalah menghasilkan aplikasi yang dapat langsung menghasilkan secara otomatis diagram sesuai dengan urutan atau tata letak diagram dan mempunyai fitur *web-enabled* serta tidak memerlukan instalasi. Dengan adanya sistem *render* otomatis model UML, memungkinkan setiap anggota tim atau *user* dapat melihat hasil kerja anggota tim lainnya atau *user* lainnya sehingga mereka dapat bekerja sama dalam proyek *software* yang sama di tempat yang berbeda. Penelitian ini juga menjelaskan setiap diagram harus memberikan posisi *triplet* atau koordinat yang dinamis dan interaktif.

Penelitian kedua menurut (Cao *et al*, 2005) menjelaskan dalam penelitiannya, aplikasi Case Tools yang dirancang adalah *desktop* aplikasi yang dilengkapi dengan fitur *website editing*, dalam aplikasi tersebut ditambahkan fitur baru di mana *user* dapat melakukan pembuatan dan *editing* diagram UML melalui *website* yang terhubung dengan aplikasi *desktop* tersebut. Untuk menghubungkan kedua aplikasi digunakan *interpreter meta-tools* yang menyediakan *server* aplikasi bersama yang didukung menggunakan arsitektur *web*.

Penelitian ketiga menurut (Mackay, Noble, Biddle, 2002), tujuan perancangan Case Tools berbasis *web* adalah menghasilkan sebuah aplikasi Case Tools yang ringan serta mudah dalam pembuatan model berbasiskan *web*. Dengan cara ini, *user* tidak perlu menginstal *plug-ins* untuk membuat UML diagram serta didukung dengan portabilitas dan *group collaboration*. Jika aplikasi diakses melalui *browser* maka tidak memerlukan *ekstension* khusus untuk dapat mengakses aplikasi. Tool UML yang digunakan dulu disebut sebagai *heavyweight tools*, di mana *tools* ini tidak berbasiskan *web* sehingga saat mengelolah data diagram, sistem *tools* ini membaca satu per satu data yang mengakibatkan *tools* tersebut berjalan lama.

Dari berbagai penelitian di atas penulis menemukan masalah yang paling banyak ditemukan dalam perancangan Case Tools adalah bagaimana membuat aplikasi yang ringan dan dapat mendukung fungsi-fungsi untuk *me-render* otomatis serta *editing* dalam diagram UML seperti *desktop* aplikasi dengan didukung fitur *collaborative*.

Rangkuman Analisis

Berdasarkan analisis berbagai penelitian sebelumnya, maka aplikasi CASE Tools berbasis *Web* ini seharusnya dirancang dengan memiliki fitur-fitur sebagai berikut: (1) *Create New Project*: fitur untuk membuat *project* baru sebagai tempat menyimpan hasil diagram UML. (2) *Create New Canvas*: fitur untuk membuat *canvas* sebagai tempat untuk meletakkan diagram yang telah penulis buat. (3) *Create Model*: fitur untuk memilih model diagram UML yang akan dibuat yang terdiri dari *class diagram*, *activity diagram*, dan *use case diagram*. (4) *Open Project*: fitur untuk membuka kembali pekerjaan *user* yang telah dibuat pada aplikasi ini. (5) *Save Project*: fitur untuk menyimpan semua pekerjaan *user* dalam aplikasi. (6) *Create Class Diagram*: fitur untuk membuat *class diagram* beserta konektor yang terdiri dari generalisasi, agregasi, komposisi, dan asosiasi. (7) *Create Activity Diagram*: fitur untuk membuat *activity diagram* dan komponennya seperti *initial* dan *final node*, *decision*, *fork/join*, serta konektor yang terdiri dari generalisasi, agregasi, komposisi, dan asosiasi. (8) *Create Use Case Diagram*: fitur untuk membuat *use case diagram* dan komponennya seperti *actor*, *use case*, dan *boundary* serta konektor asosiasi. (9) *Add Relation Use Case*: fitur untuk membuat hubungan atau relasi untuk diagram *use case* berupa konektor diagram asosiasi. (10) *Add Circle Use Case*: fitur untuk membuat *circle* atau lingkaran yang merupakan bagian dari *use case diagram*. (11) *Add Attribute Class*: fitur untuk membuat dan menambahkan *attribute* pada *class diagram*. (12) *Add Operation Class*: fitur untuk membuat dan menambahkan *operation* pada *class diagram*. (13) *Add Relation Class*: fitur untuk membuat hubungan atau relasi pada *class diagram* dalam bentuk konektor diagram generalisasi, agregasi, komposisi, dan asosiasi. (14) *Add New Item Activity*: fitur untuk membuat *item* untuk *activity diagram* yang menjelaskan aktivitas diagram tersebut yang menggambarkan aktivitas dari proyek. (15) *Add Relationship Activity*: membuat hubungan atau relasi untuk *activity diagram* dalam bentuk konektor diagram generalisasi, agregasi, komposisi, dan asosiasi.

Analisis Aplikasi Sejenis

Pada bagian ini akan dijelaskan hasil analisis penulis tentang perbandingan dengan aplikasi sejenis yang sudah ada dengan yang penulis buat. Penulis mengambil tiga contoh aplikasi berbasis *desktop* sebagai pembanding. Karena aplikasi CASE Tools berbasis *web* jarang ditemui, penulis

memakai aplikasi *desktop* sebagai acuan untuk membuat CASE Tools dengan fitur yang lengkap dan baik. Salah satu produk yang sangat terkenal adalah IBM Rational Software Architect.

IBM Rational Software Architect adalah salah satu dari aplikasi CASE Tools berbasis *desktop* yang digunakan untuk memodelkan UML, mendokumentasi, serta mengelompokkan fase perancangan perangkat lunak berorientasi objek.

Kelebihan IBM Rational Software Architect antara lain: (1) Memiliki fitur yang lengkap seperti *business modeling* dan analisis, *data modeling*, *reporting*. (2) Dapat membuat dokumentasi dalam bentuk *reporting*. (3) Mendukung fitur *collaboration* yang mendukung untuk membuat *multiple diagram* dalam satu *project*. (4) Mendukung pemodelan *database*.

Sedangkan kelemahan yang dimiliki oleh IBM Rational Software Architect antara lain: (1) Kompleks untuk digunakan oleh *user* awam, pada umumnya aplikasi ini digunakan oleh *user* yang sudah berpengalaman dalam membuat pemodelan *software*. (2) Fitur terlalu banyak sehingga membuat *user* bingung untuk menemukan fitur yang diinginkan. (3) Memerlukan spesifikasi khusus yang harus dipenuhi untuk meng-*install* aplikasi *software* ini.

Hasil Rancangan

Berdasarkan analisis di atas dapat ditentukan kriteria yang akan menjadi dasar pengamatan terhadap aplikasi tersebut, seperti fitur *Bussiness Modeling*, *Save Project*, *Multiple Diagram*, *Data Modeling*, *Installation Needed*, *Easy User Interface*, dan *Web Supporting*. Dari aplikasi tersebut memiliki kesamaan fitur yaitu *Installation Needed*.

Instalasi yang dimaksud merupakan hal yang sangat penting karena jika ingin menggunakan aplikasi tersebut maka diwajibkan menginstal aplikasi tersebut pada PC atau perangkat pengguna. Tidak semua instalasi *software* atau aplikasi dapat dilakukan pada semua PC ataupun peringkat, karena untuk melakukan instalasi diperlukan spesifikasi minimum dan dibutuhkan *platform* tertentu supaya proses instalasi dapat berjalan dengan lancar. Jika spesifikasi minimum dan *platform* tidak terpenuhi, proses instalasi tidak akan dapat dilakukan sehingga *user* tidak dapat memakai perangkat lunak atau aplikasi tersebut.

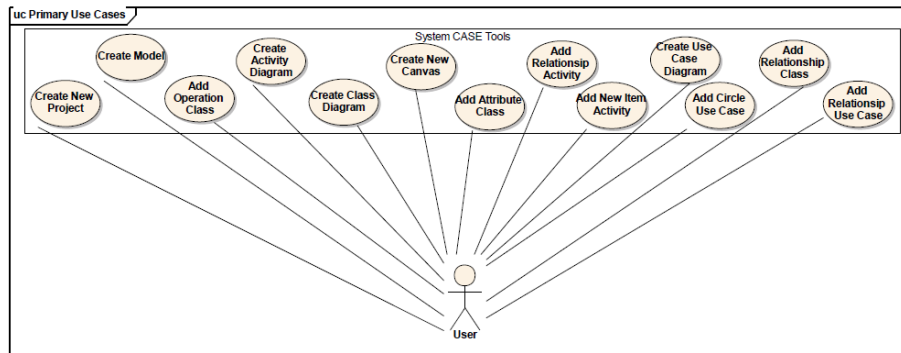
Berdasarkan hasil analisis di atas, maka akan dibuat aplikasi berbasis *Web* sehingga tidak memerlukan instalasi dan dapat langsung digunakan hanya dengan mengakses *website* serta tidak memerlukan spesifikasi minimum dan dapat berjalan pada semua *platform*. Aplikasi yang dirancang memiliki fitur-fitur, antara lain: (1) *Save Project*: fitur ini akan menyimpan semua hasil pekerjaan *user* dalam sebuah *project*, dan ketika *user* ingin membuka *project*-nya kembali maka aplikasi ini akan me-*reload project* tersebut dan akan menampilkan *state* terakhir *user* saat menyimpan pekerjaannya. (2) *Multiple Diagram*: fitur ini akan memudahkan *user* untuk membuat berbagai jenis diagram dalam satu *project*. (3) *Web Supporting*: Dengan fitur ini aplikasi tersebut tidak akan memerlukan instalasi apapun sehingga *user* dapat langsung mengakses dan menggunakan aplikasi, karena aplikasi ini dirancang agar dapat berjalan pada basis *website*. (4) *Easy User Interface*: Fitur yang memudahkan *user* dalam menggunakan aplikasi, tampilan atau *interface* akan dirancang sedemikian rupa agar *user* tidak bingung dalam menggunakan aplikasi.

Rancangan Aplikasi

Aplikasi CASE Tools: Diagram Use Case

Aplikasi terdiri dari tiga bagian utama antara lain *New Project* berfungsi untuk mengarahkan pengembang agar dapat membuat *project* baru pada aplikasi, akan muncul sebuah jendela baru untuk *user* meng-*input* nama *project* dan direktori tempat penyimpanannya, setelah *user* meng-*create project*

maka *user* akan diarahkan ke layar Case Tools UML dimana *user* dapat mengedit dan menambahkan diagram. *Open Project* berfungsi untuk membuka *project* yang sudah ada, dan aplikasi akan *reload* dan menampilkan kembali pekerjaan *user* pada saat terakhir kali *save*. *Collaboration* berfungsi agar pengembang dapat bekerja dalam satu *project* yang sama dengan pengembang lain, pengembang dapat saling menambahkan dan mengurangi diagram dalam pekerjaan mereka dan aplikasi akan memutakhirkan secara langsung tanpa perlu *me-refresh* halaman *website*.



Gambar 1 Diagram Use Case Aplikasi CASE Tools

Rancangan aplikasi Case Tools adalah dimulai dengan menampilkan halaman *web* di mana pengembang dapat mengedit dan menambahkan diagram UML pada proyek yang sedang dikerjakan. Desain layar Case Tools terdiri dari lima bagian yaitu *menu* yang berfungsi agar pengembang dapat membuat proyek baru, membuka proyek yang sudah ada, serta dapat menyimpan proyek yang telah dibuat. Bagian *menu* ini akan membantu pengembang untuk *men-setting* tampilan layar Case Tools. *Project* akan menampilkan semua pekerjaan pengembang berupa *folder project* dan diagram beserta nama *project* dan nama diagram lalu menampilkannya dalam bentuk *treemenu*. *Pallette* berisi tipe diagram yang disediakan oleh aplikasi dan pengguna dapat menggunakannya dengan *men-drag* ke dalam *canvas*. *Properties* menampilkan *attribute* yang terdapat pada diagram. *Canvas* adalah tempat semua diagram yang dibuat oleh *user* diletakkan.

Create New Project berfungsi agar *user* dapat membuat *project* baru, setiap ingin membuat *project* baru maka akan muncul jendela baru yang akan meminta *user* untuk meng-*input* nama *project* dan menentukan ekstensi untuk menyimpan *project* tersebut. Setelah selesai meng-*input* maka *user* dapat memulai membuat diagram dan semua *project user* akan ditampilkan pada bagian *Project*.

Create New Model berfungsi agar *user* memilih tipe diagram apa yang akan dibuat. Tipe diagram pada aplikasi ini terdiri dari *class diagram*, *use case diagram*, dan *activity diagram*. Jendela akan muncul setelah pengembang selesai meng-*input* nama dan direktori penyimpanan *project*.

Pallette Diagram berfungsi menampilkan tipe diagram dan komponen dari masing-masing tipe diagram, untuk memakai tipe diagram dan komponennya maka *user* harus *men-drag* tipe diagram tersebut ke dalam *canvas* yang terdapat pada bagian tengah layar Case Tools. Setelah itu aplikasi akan menampilkan diagram sesuai dengan tipe yang di-*drag* oleh *user*.

User dapat menambahkan *attributes* pada diagram, saat *user* *men-drag* tipe diagram dari *pallette* aplikasi juga akan memunculkan jendela baru agar *user* dapat menginput *attributes* ataupun *operation* pada diagram. Nilai dari masukan pengguna akan ditampilkan pada diagram dan juga pada bagian *properties*. Pengguna juga dapat mengedit nilai tersebut dengan menggantinya pada bagian *properties*.

Add Diagram Type berfungsi untuk menambahkan tipe diagram baru pada *project user*. *User* dapat menambah diagram dengan meng-klik kanan pada nama *project* yang *user* buat pada bagian *Project*, setelah itu akan muncul pilihan untuk memilih tipe diagram. Pada bagian *Project* akan ter-update diagram baru dan aplikasi akan meng-create sebuah *canvas* baru.

Pengguna dapat meng-add *conector* pada diagram yang fungsinya untuk menjelaskan hubungan diagram yang satu dengan diagram lainnya. Setiap diagram memiliki *connector* masing-masing dengan jenis berbeda, pengguna dapat membuat *connector* pada diagram dengan menghubungkan kedua lingkaran kecil yang terdapat pada diagram dan menarik garis pada lingkaran tersebut untuk dihubungkan ke diagram lain. Lingkaran tersebut akan muncul jika *user* meng-klik pada diagram.

Aplikasi CASE Tools: Diagram Class

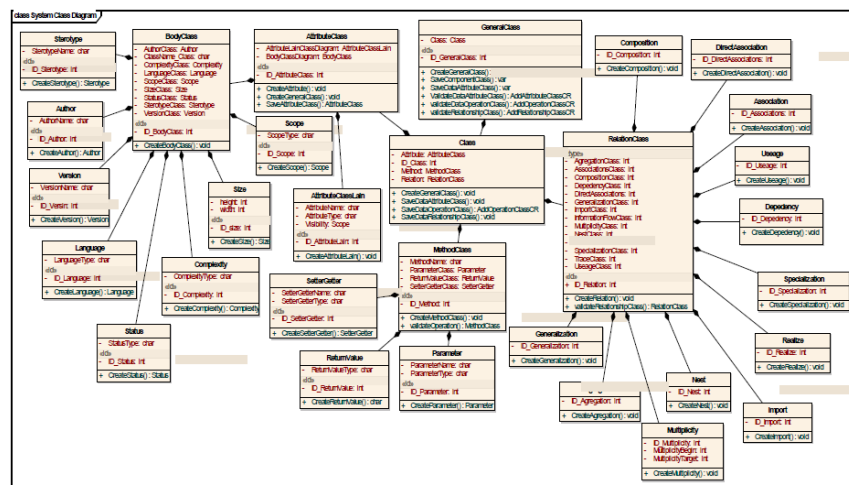
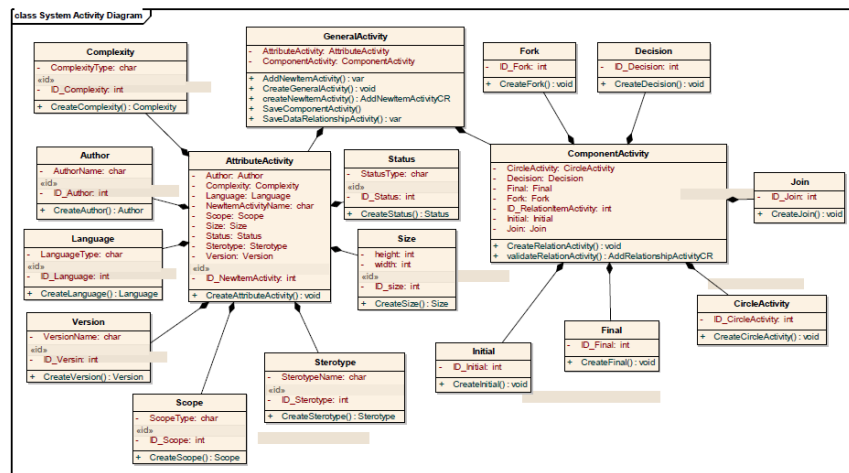
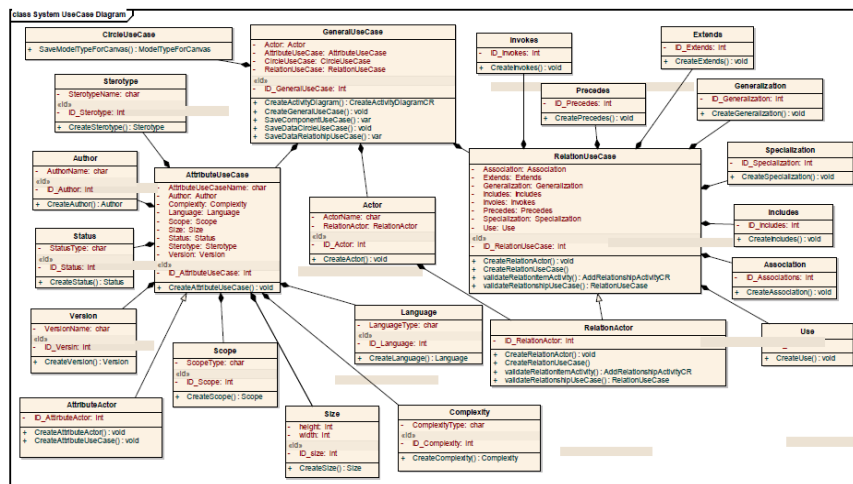
Untuk penyimpanan hasil permodelan adalah di dalam bentuk *database*. Setiap objek yang ada direpresentasikan dalam bentuk *class*, sebagai bentuk baru dan lebih komprehensif dari bentuk yang sebelumnya disebut *entity*. Rancangan dasar dari objek adalah tipe objek, yang mewakili kelompok objek dengan sifat yang sama. Sebuah tipe objek memiliki eksistensi independen terhadap keberadaan fisik benda terhadap keberadaan konseptual (atau abstraksi). Setiap objek dapat diidentifikasi secara unik dari suatu *entity* yang disebut hanya sebagai suatu kejadian entitas. Sebuah *database* biasanya berisi banyak jenis objek kelas yang berbeda. Pada Gambar 2, Gambar 3, dan gambar 4 merupakan representasi objek yang diperlukan di dalam setiap pembuatan diagram setiap aplikasi bisnis, yaitu dalam pembuatan diagram *use case*, diagram *class*, dan pembuatan diagram *activity*. Setiap jenis entitas ditampilkan sebagai persegi panjang diberi label dengan nama entitas, yang biasanya kata benda tunggal yang merupakan identitas dari objek bersangkutan.

Daftar kelas objek untuk mendukung aplikasi sesuai dengan gambar 2, gambar 3, dan gambar 4 dan sesuai dengan rincian penjelasan pada Tabel 1.

Tabel 1 Daftar Kelas Objek untuk mendukung Aplikasi CASE Tools

No	Nama Kelas Objek	Atribut
1	modelType	NameModelType : CHAR (36), IdModelType : CHAR (36)*
2	general_Model	ModelName : VARCHAR (100) IdGeneralModel: CHAR (36)* IdModelType : CHAR (36)#
3	general_Canvas	CanvasName : CHAR (36), IdGeneralCanvas : CHAR (36)*
4	general_Project	ProjectName : VARCHAR (100) IdGeneralProject: CHAR (36)*, IdGeneralCanvas : CHAR (36)#
5	typeClass	typeclassID: CHAR(36)*, typeclassName: CHAR(36), idMethod: CHAR(36)#, idAttribute: CHAR(36)#, RelationClassID: CHAR(36)#
6	general_Class	classname: VARCHAR(100), idgeneralclass: CHAR(36)*, typeclassID: CHAR(36)#
7	attribute_class	Name: CHAR(36), idAttribute: CHAR(36)*
8	general_Activity	Activity_Name: VARCHAR(100), ID_generalActivity: CHAR(36)*, AttributeActivityID : CHAR(36)#, ComponentActivityID : CHAR(36)#
9	AttributeActivity	AttributeActivityName: CHAR(36), AttributeActivityID : CHAR(36)*
10	MethodClass	Name: CHAR(36), idMethod: CHAR(36)*
11	RelationClass	Name: CHAR(36),RelationClassID: CHAR(36)*
12	componentActivity	ComponentActivityName : CHAR (36), ComponentActivityID : CHAR (36)*
13	general_UseCase	UseCase_Name : VARCHAR(100), ID_GeneralUseCase : CHAR(36)*, CircleUseCaseID : CHAR(36)#, RelationUseCaseID : CHAR(36)#
14	circle_UseCase	CircleName : CHAR(36), CircleUseCaseID : CHAR(36)*

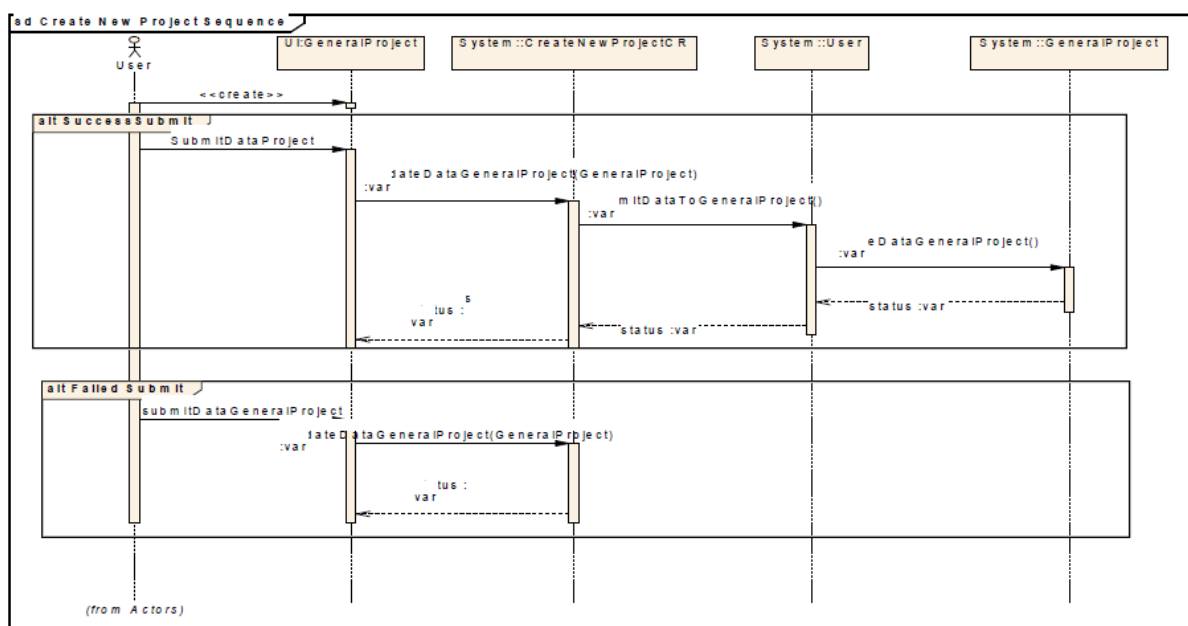
Catatan: * primary key, # foreign key



Aplikasi CASE Tools: Diagram Sequence

Produk selanjutnya dari rancangan CASE Tools, adalah berupa diagram *sequence*. Model diagram *sequence* merepresantasikan interaksi antar objek sepanjang waktu. Dalam diagram ditunjukkan objek dan pesan yang bergerak dari suatu objek dan ke objek yang lain sesuai yang ditunjukkan dalam diagram *use case*. Dalam diagram *sequence object* dan *actor* ditampilkan sebagai kolom, dengan *vertical lifelines* menunjukkan waktu hidup pada garis waktu. Sedangkan aktivitas ditunjukkan dengan notasi kotak persegi panjang pada *horizontal life lines*.

Pada gambar 5, diperlihatkan salah satu diagram *sequence*, namun karena jumlah *diagram sequence* yang terlalu banyak jadi tidak bisa dimuat dalam tulisan ini. Namun terdapat dalam referensi pada artikel ini, dengan judul dan penulis yang sama.



Gambar 5 Sequence Diagram dari Create New Project

Hasil Evaluasi

Pada sub-bab ini dilakukan evaluasi terhadap *user* dengan menggunakan kuesioner dengan tujuan untuk mengetahui apakah aplikasi dengan *platform website* ini lebih dibutuhkan dari pada aplikasi CASE Tools berbasis *desktop* pada umumnya. Target *responder* penulis adalah mahasiswa Bina Nusantara jurusan Teknik Informatika angkatan 2009, alasannya karena aplikasi ini dirancang dengan tujuan agar dapat memenuhi kebutuhan standar mahasiswa dalam pembuatan permodelan diagram UML.

Pertanyaan-pertanyaan diajukan untuk mengetahui persentase pilihan *responder* untuk aplikasi *open source* atau dapat digunakan tanpa harus membayar *license*, karena pada umumnya aplikasi CASE Tools banyak yang berbayar atau harus menggunakan *license*. Hasilnya 95% mengatakan ingin, dan 5% lainnya tidak. Dari hasil evaluasi kuesioner terhadap *responder* maka dapat diketahui bahwa aplikasi CASE Tools berbasis *website* ini sangat dibutuhkan terutama untuk kebutuhan membuat pemodelan dengan UML yang mudah dalam penggunaan interface aplikasi dan *open source*, tidak memerlukan instalasi, dan memiliki fitur *save project*.

SIMPULAN

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan bahwa dari perancangan aplikasi CASE Tools berbasis *Web* ini yang telah dikembangkan menyangkut beberapa hal: pertama, aplikasi yang dirancang berbasis *Web* ini tidak memerlukan instalasi dan spesifikasi yang besar dalam menggunakan atau menjalankan aplikasi. Selanjutnya, dengan berbasis *Web*, maka semua data *project* UML yang dibuat oleh *user* pada aplikasi ini akan disimpan secara *online*, dan *user* dapat membuka kembali proyek tersebut. Hal ini memudahkan *user* dalam menyimpan dan membuka kembali semua pekerjaannya karena semua data telah disimpan pada aplikasi. Aplikasi ini dirancang dengan standarisasi tampilan *interface* dan fitur-fitur CASE Tools yang biasanya terdapat pada *desktop* aplikasi sehingga memudahkan bagi *user* sudah terbiasa menggunakan aplikasi CASE Tools berbasis *desktop*. Aplikasi ini dirancang agar dapat berjalan secara ringan dan cepat dengan fitur-fitur yang biasanya hanya terdapat aplikasi *desktop*. *User* dapat menggunakan aplikasi CASE Tools berbasis *Web* ini seperti menggunakan aplikasi *desktop*.

Berdasarkan kesimpulan di atas, yang diperoleh dengan keterbatasan yang terdapat pada perancangan aplikasi ini, maka saran yang dapat dipertimbangkan untuk pengembangan aplikasi selanjutnya adalah: (1) Pengembangan aplikasi di mana tidak hanya untuk pemodelan UML saja, tapi juga untuk pemodelan *database* dan proses bisnis untuk pembuatan *software*. (2) Menambahkan fitur *online collaboration* di mana sesama *user* dapat mengerjakan satu project secara bersamaan.

DAFTAR PUSTAKA

- Cao, S., Grundy, J., Hosking, J., Stoeckle, H., Tempero, E., & Zhu, N. (2005). Generating Web-based User Interfaces for Diagramming Tools. *Department of Computer Science and Department of Electrical and Computer Engineering University of Auckland*, 1-10.
- Cervone, H. (2011). *Understanding agile project management methods using Scrum*. (OCLC Systems and Services.
- Mackay, D., Noble, J., Biddle, R. (2002). A Lightweight Web-Based Case Tool for UML Class Diagrams. *Victoria University Of Wellington*, 1-7.
- Palaniappan, S., & Ling, L. (2008). Web-Based CASE Tool for Automated Rendering of UML. *IJCSNS International Journal 146 of Computer Science and Network Security*, 1-7.
- Sutherland, J. (2010). *Jeff Sutherland's Scrum Handbook*. Somerville: The Scrum Training Institute