

## PENGEMBANGAN PORTAL ANAK SEKOLAH MINGGU SEBAGAI BAGIAN DARI SISTEM INFORMASI TERPADU GEREJA MASEHI INJILI DI MINAHASA

Stanley Karouw<sup>1</sup>, Benny Narasiang<sup>2</sup>

(1),(2) Universitas Sam Ratulangi Manado, stanley.karouw@unsrat.ac.id

### Abstract

Gereja perlu memanfaatkan internet untuk kepentingan penyebaran Injil dan penguatan iman warga gereja, terlebih anak-anak. Salah satu diantaranya dengan membuat konten yang terkait pemahaman Alkitab dan ajaran, yang membangun iman dan percaya warga gereja, khususnya generasi yang akan datang. Teknologi internet, seperti Web 2.0 memungkinkan Gereja untuk mengintensifkan penyebaran konten dalam waktu singkat dengan biaya yang relative rendah. Aplikasi Portal Sekolah Minggu berbasis Web 2.0 merupakan reasonable solution untuk mendukung tujuan penyebaran konten terkait pemahaman Alkitab untuk Guru dan Anak Sekolah Minggu. Aplikasi ini pun efektif meningkatkan interaksi antar Guru – Anak dan Anak – Anak dalam proses pembelajaran Sekolah Minggu. Metodologi Agile Unified Proses (AUP) dapat mengembangkan aplikasi perangkat lunak yang berorientasi obyek, bersifat architecture-centric dan berbasis Web yang dikerjakan secara iterative, dalam waktu yang relative singkat.

**Kata kunci:** Aplikasi, Web, Agile, UML 2.0

### 1. PENDAHULUAN

#### 1.1 LATAR BELAKANG

Teknologi Informasi dan Komunikasi (TIK) semakin meluas perkembangannya. Web 2.0 sebagai salah satu produk TIK semakin dikenal dan digunakan oleh masyarakat luas, sebagai alat bantu untuk meningkatkan efisiensi kerja. Gereja sebagai organisasi pun mulai memanfaatkan Web 2.0 sebagai salah satu alat bantu efisiensi pengelolaan administrasi organisasi dan sebagai sarana penyampaian dan penyebarluasan berita Injil. Gereja perlu memanfaatkan internet dan teknologi Web 2.0 untuk kepentingan penyebaran Injil dan penguatan iman warga gereja, terlebih anak-anak. Salah satu diantaranya dengan membuat konten – konten yang terkait pemahaman Firman dan ajaran, termasuk konten-konten yang membangun iman dan percaya warga gereja, khususnya generasi yang akan datang.

#### 1.2 PERMASALAHAN PENELITIAN

Masalah yang ingin dijawab dalam penelitian ini adalah: 1) bagaimana mengembangkan sebuah aplikasi portal yang dapat menjadi Alat Peraga untuk mendukung proses pembelajaran Anak Sekolah Minggu? 2) Bagaimana menggunakan metodologi AUP dalam mendukung proses pengembangan aplikasi yang berorientasi obyek, bersifat architecture-centric, berbasis web dan dalam waktu singkat?

### 2. LANDASAN TEORI

#### 2.1 Model Proses Daur Hidup Perangkat Lunak

Model proses daur hidup perangkat lunak, dikemukakan oleh Schach<sup>[1]</sup>, merupakan tahapan pengembangan perangkat lunak ideal. Model ini menganggap perangkat lunak sebagai produk yang dihasilkan dalam urutan tahapan tertentu secara ideal. Tahapan berurutan tersebut adalah: 1) Memulai dari *scratch* (yakni memulai dari tidak ada); 2) Tahap pendefinisian *requirements* (atau kebutuhan); 3) Tahap Analysis; 4) Tahap Perancangan; 5) Tahap Implementasi.

Sommerville<sup>[2]</sup> mengemukakan empat tahapan fundamental dalam model proses perangkat lunak, yakni; 1) *Software specification* (proses pendefinisian kebutuhan perangkat lunak); 2) *Software design and implementation* (mengembangkan perangkat lunak yang sesuai dengan persyaratan user); 3) *Software validation* (perangkat lunak yang dihasilkan harus disesuaikan kembali menurut keinginan user); 4) *Software evolution* (perangkat lunak dikembangkan terus untuk memenuhi kebutuhan user yang bertambah). Pressman<sup>[3]</sup> mengusulkan suatu *generic process framework* perangkat lunak, dengan tahapan sebagai berikut: 1) Komunikasi; 2) Perencanaan; 3) Pemodelan; 4) Konstruksi; 5) Implementasi.

Dennis, Wixom dan Tegarden<sup>[4]</sup> mengemukakan model proses yang disebut *Sistem Development Life Cycle* (disingkat SDLC) dengan tahapan berikut: 1) Perencanaan, 2) Analisis, 3) Perancangan, 4) Implementasi. Tahapan ini serupa dengan yang dikemukakan oleh Bentley dan Whitten<sup>[5]</sup>, yakni: 1) *Sistem Initiation*; 2) *Sistem Analysis*; 3) *Sistem Design* dan 4) *Sistem Implementation*. Sedangkan Kendall dan Kendall<sup>[6]</sup> mengusulkan 7 (tujuh) tahapan dalam SDLC, yakni: 1) Identifikasi permasalahan, kesempatan dan tujuan; 2) Penentuan persyaratan informasi pengguna; 3) Analisa kebutuhan sistem; 4) Perancangan sistem yang telah direkomendasi; 5) Pengembangan dan dokumentasi perangkat lunak; 6) Menguji sistem; 7) Implementasi dan Evaluasi sistem.

Terkait dengan model proses perangkat lunak, maka *Software Engineering Institute – Carnegie Mellon (SEI)*<sup>[7]</sup> mengeluarkan *framework* Standar Ukuran Kematangan yang disebut *CMMI for Development (CMMI – DEV)*. Model *CMMI® (Capability Maturity Model® Integration)* merupakan kumpulan *best practices* yang membantu setiap organisasi untuk mengembangkan proses pengembangan perangkat lunak. Model ini dikembangkan dari kalangan industry, pemerintahan dan akademisi pada SEI. Model proses yang disebut CMMI-DEV, menyediakan kumpulan panduan

lengkap terkait pengembangan layanan dan produk perangkat lunak.

Menurut Schach<sup>[1]</sup>, model daur hidup perangkat lunak, secara ideal berbeda dengan praktek dikarenakan dua hal:

- 1) praktisi perangkat lunak adalah manusia, sehingga cenderung untuk membuat kesalahan;
- 2) kebutuhan pengguna cenderung mengalami perubahan saat perangkat lunak sementara dikembangkan.

### 2.2 Model Daur Hidup PAUS

Model daur hidup *PAUS ver 1.1 (Pusikom Agile Unified Process version 1.1)* adalah sebuah kerangka kerja pengembangan perangkat lunak yang dikembangkan oleh Enterprise Computing Lab – Fakultas Ilmu Komputer Universitas Indonesia. PAUS merupakan serangkaian disiplin tertentu dengan pembagian tugas dan tanggung jawab komponen organisasi pengembang software. PAUS menekankan pada karakteristik agility dan berorientasi obyek. Karakteristik agility menekankan pada keterlibatan pengguna dalam setiap proses pengembangan perangkat lunak. Orientasi obyek pada PAUS menekankan implementasi guna-ulang komponen aplikasi yang dikembangkan. PAUS dapat membantu tim pengembang untuk bekerja lebih efisien dalam menghasilkan produk software yang berkualitas, yakni memenuhi kebutuhan pengguna secara tepat waktu dan tepat anggaran. PAUS dikembangkan dengan mengadaptasi pendekatan Agile dari Ambysoft(c) dan Rational Unified Process (RUP).

### 2.3 Unified Modelling Language (UML)

UML adalah singkatan dari *Unified Modeling Language*, yaitu suatu notasi pemodelan aplikasi perangkat lunak. Schach<sup>[1]</sup> menegaskan bahwa UML merupakan bahasa bukan metode. Sebagai bahasa, UML digunakan untuk mendeskripsikan perangkat lunak yang dikembangkan dengan berbagai paradigma pengembangan perangkat lunak dan metodologi. Pendapat Schach<sup>[1]</sup> didukung oleh Sommerville<sup>[2]</sup> dan Pressman<sup>[3]</sup>.

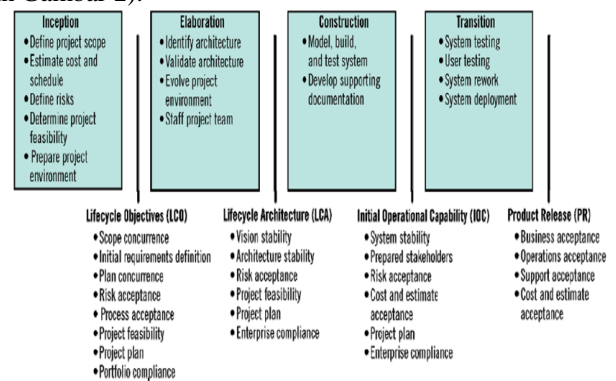
Dennis, Wixom dan Tegarden<sup>[4]</sup> mendukung pendapat bahwa UML merupakan kumpulan standar pemodelan dengan menggunakan diagram, dimana UML bertujuan untuk menyediakan kosa-kata dari paradigma pengembangan sistem berorientasi obyek guna memodelkan semua tahapan dari daur hidup pengembangan perangkat lunak. Bentley dan Whitten<sup>[5]</sup>, mendukung pemahaman bahwa UML merupakan kumpulan alat pemodelan yang disepakati bersama untuk menjelaskan sistem perangkat lunak. Hal serupa dikemukakan oleh Kendall dan Kendall<sup>[6]</sup>.

Fowler<sup>[8]</sup> memberikan definisi yang sederhana bahwa UML merupakan kumpulan notasi grafis, yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek. UML merupakan standar yang relatif terbuka yang diatur oleh *Object Management Group (OMG)*, sebuah konsorsium terbuka. OMG berfungsi untuk membuat standar-standar yang mendukung interoperabilitas sistem yang berorientasi objek. Versi terakhir dari UML adalah UML ver 2.0<sup>[9]</sup>.

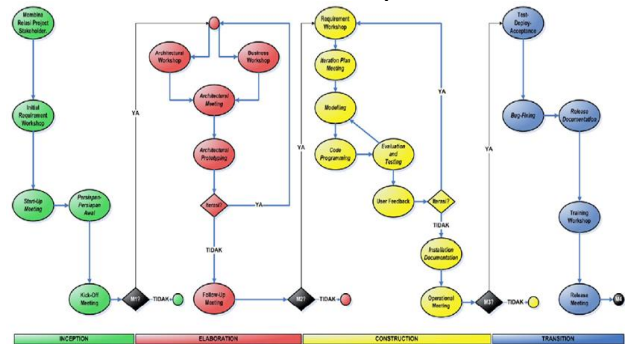
Menurut Kruchten<sup>[10]</sup>, UML adalah bahasa grafis untuk *visualizing, specifying, constructing and documenting* setiap artifak dari sistem perangkat lunak. UML mendukung *The 4+1 View Model of Architecture*, yakni 1) *The Logical View*, 2) *The Implementation View*, 3) *The Process View* dan 4) *The Deployment View* ditambah dengan 5) *The Use Case View*. Model merupakan representasi lengkap dari sistem perangkat lunak, sedangkan arsitektur merupakan fokus pandangan pada bagian-bagian tertentu dari sistem perangkat lunak. Atau dapat dikatakan arsitektur sistem merupakan cetak-biru aplikasi. Keterhubungan model dan arsitektur sistem perangkat lunak, digambarkan oleh UML.

### 3. METODOLOGI PEMECAHAN MASALAH

Tahapan pemecahan masalah dengan metodologi AUP, mengikuti langkah-langkah yang dikeluarkan oleh Pusikom Universitas Indonesia<sup>[11]</sup>. Panduan Agile UP ini mengacu pada metodologi yang dibuat oleh *Ambysoft Inc*<sup>[12]</sup> (lihat Gambar 1 dan Gambar 2).



Gambar. 1 Tahapan AUP



Gambar 2. Aktivitas Proses AUP

Tahapan analisa dan perancangan AUP adalah sebagai berikut:

- 1) *Inception*, dengan aktivitas mendefinisikan *project scope*, mengestimasi biaya dan penjadwalan, mendefinisikan resiko, membuat kelayakan proyek dan mempersiapkan lingkungan pengerjaan proyek (tim, tempat kerja, instalasi, dan sebagainya). Proses iterasi dilakukan satu kali. Artifak yang dihasilkan diantaranya adalah dokumen *Stakeholder Request (STRQ)*, *Vision*, dokumen *Supplementary Specification* dan dokumen *Software Project Plan* yang berisi estimasi perangkat lunak, kelayakan Finansial (misalnya dengan ROI dan NPV), *Workplan*, *Scenario Test Plan* dan Daftar Resiko.

2) *Elaboration*, dengan aktivitas mengidentifikasi dan validasi arsitektur aplikasi. Proses iterasi dapat dilakukan satu sampai dua kali. Artifacts yang dihasilkan adalah *Software Requirements Specification* dan *Software Architecture Document*, serta artifacts fase Inception yang telah diperbaharui.

3) *Construction*, dengan aktivitas memodelkan, membangun dan menguji sistem aplikasi serta membuat dokumentasi pendukung. Proses iterasi dapat dilakukan dua hingga delapan kali. Artifacts yang dihasilkan adalah *Source Code Document*, *Test Report* dan semua artifacts fase Elaboration yang telah diperbaharui (SRS dan SAD).

4) *Transition*, dengan aktivitas menguji sistem (*integration sistem* dan *user testing*), mereview kembali sistem aplikasi dan menginstalasi sistem aplikasi. Proses iterasi dapat dilakukan satu hingga dua kali. Artifacts yang dihasilkan adalah, Panduan Instalasi dan Panduan Pengguna, Dokumen Pelatihan dan semua dokumen fase Elaboration dan Construction yang telah diperbaharui.

Panduan Agile UP dari Pusilkom UI juga memberikan *best practice* dalam melakukan setiap aktivitas di setiap fase. Panduan ini juga membedakan artifacts dokumen yang dihasilkan, sebagai artifacts utama, artifacts pendukung dan artifacts input dan artifacts output. Panduan ini juga memberikan LCO (*Lifecycle Objective*) berupa dokumen dan presentasi dari setiap fase, sebagai target yang harus dicapai sebelum melanjutkan ke fase yang selanjutnya. Untuk kepentingan penulisan paper ini, maka penulis akan membatasi artifacts yang akan ditampilkan.

**4. PEMBAHASAN**

**4.1 Fase Inception**

**4.1.1 Tujuan Aktivitas Proses**

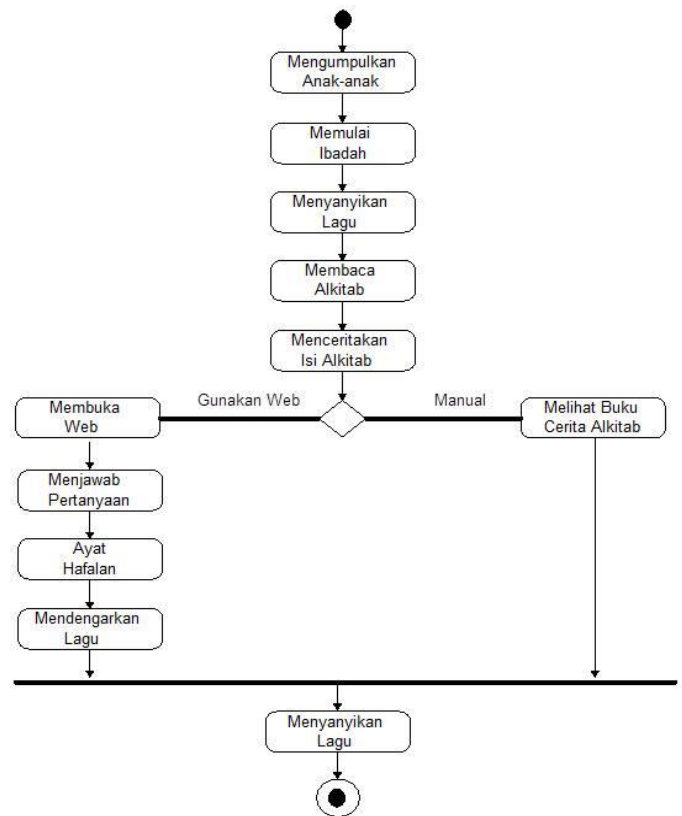
Target utama fase inception adalah memahami cakupan dan tujuan proyek serta memperoleh cukup informasi yang bisa mengkonfirmasi bahwa kita harus jalan terus (atau sebaliknya mengapa tidak perlu diteruskan). Lima tujuan dasar fase inception adalah: a) Memahami apa yang hendak dibangun. Menentukan visi, cakupan sistem dan batasannya; b) Mengidentifikasi fungsionalitas sistem. Memutuskan use case mana yang paling critical; c) Menentukan setidaknya satu solusi yang paling mungkin. Identifikasi setidaknya sebuah konfigurasi yang menjadi kandidat arsitektur sistem; d) Memahami ongkos, jadwal dan resiko yang berkaitan dengan proyek; e) Menentukan proses apa yang harus diikuti dan tools mana yang akan digunakan.

**4.1.2 Artifacts LCO Fase Inception**

Dokumen STRQ dan Vision merangkum tujuan pertama, kedua dan ketiga dari fase inception. Berikut adalah gambar artifacts terkait:

**a. Proses Bisnis Sekolah Minggu.**

Proses bisnis Sekolah Minggu ditunjukkan pada Gambar 3 berikut ini. Proses bisnis merupakan rangkaian aktivitas yang dikerjakan dalam Sekolah Minggu. Pada Gambar proses bisnis, ditunjukkan setelah pada aktivitas Menceritakan Isi Alkitab, maka dapat digunakan teknologi portal web.



**Gambar. 3** Proses Bisnis Sekolah Minggu

b. Problem Statement dan Key Stakeholders Need  
Permasalahan dan kebutuhan key stakeholders dirangkum pada Tabel berikut:

**Tabel 1.** Permasalahan Proses Bisnis

The problem of	Anak Sekolah Minggu tidak tertarik atau berkurang rasa tertariknya karena gaya mengajar yang monoton dan berulang-ulang. Guru Sekolah Minggu sulit mendapatkan bahan ajar
affects	Kualitas kegiatan Sekolah Minggu
the impact of which is	Tidak dapat membawakan Cerita Alkitab sesuai dengan program Tidak dapat membuat anak Sekolah Minggu memahami dengan baik pesan dari Cerita Alkitab
a successful solution would be	Portal Web Sekolah Minggu yang interaktif dan lengkap berisi bahan ajar

**c. Fungsionalitas Utama**

Fungsionalitas utama dibedakan menjadi Functional Requirements dan Non Functional Requirements yang dirangkum pada Tabel berikut:

**Tabel 2. Fungsionalitas Utama**

Kebutuhan Fungsional	
1. Melakukan Login	1.1 Login Administrator 1.2 User Viewing
2. Mengelola Data	2.1 Admin dapat melakukan input, edit, update dan delete Daftar Soal 2.2 Admin dapat melakukan input, edit, update dan delete Daftar Ayat Hafalan 2.3 Admin dapat melakukan input, edit, update dan delete Daftar Lagu 2.4 Admin dapat melakukan input, edit, update dan delete Daftar Cerita
3. Sharing Data	3.1 Sistem dapat sharing data ke sosial media
Kebutuhan Non Fungsional	
1. Kebutuhan Operasional	1.1 Aplikasi dapat dijalankan pada Sistem Operasi Xp dan Windows 7 1.2 Bahasa yang digunakan adalah Bahasa Indonesia
2. Kebutuhan Performance	2.1 Respons time halaman 1-5 detik

Dokumen *Software Project Plan* merangkum tujuan keempat dan kelima dari fase inception. Berikut adalah penggalan artifact terkait:

#### d. Estimasi Software

Estimasi perangkat lunak, dihitung dengan menggunakan teknik Function Point Analysis. Ringkasan hasil perhitungannya adalah:

**Tabel 3. Estimasi Aplikasi Portal Sekolah Minggu**

Estimasi Aplikasi Portal Sekolah Minggu	
Total Adjusted Function Point	160.13
Lines of Code (LOC)	2400
Effort (in person-months)	3
Estimate Time Required	3

#### e. Kelayakan Finansial

Dokumen kelayakan fungsional dapat dilihat cuplikannya pada Gambar 4 dibawah ini. Estimasi kelayakan fungsional dihitung dengan menggunakan Net Present Value (NPV), Return on Investment (ROI) dan Break-event Point (BEP).

#### f. Daftar Resiko

Berikut ditampilkan dua resiko yang most-likely dapat terjadi dalam pengembangan aplikasi ini

**Tabel 4. Daftar Resiko #1**

<b>Risk#1</b>	Feature Creep
<b>Likelihood of risk</b>	Tinggi ( <i>high probability of risk</i> )
<b>Potential impact on the project</b>	Bertambahnya fitur yang harus dikerjakan oleh tim pengembang
<b>Ways to address this risk</b>	Finalisasi fitur aplikasi pada dokumen tertulis; Melakukan dokumentasi manajemen perubahan

	2012	2013	2014	Total
Mengurangi biaya pembuatan alat peraga	1.835.000	1.835.000	1.835.000	
Mengurangi biaya fotocopy	720.000	720.000	720.000	
Total Benefit	2.555.000	2.555.000	2.555.000	
PV of Benefits	238.785,046	238.785,046	238.785,046	716.355,138
Software Licenses	0	0	0	
Server software	0	0	0	
Development Labor	270.000	0	0	
Total Development Costs	270.000	0	0	
Hardware	0	0	0	
Software	0	0	0	
Operational Labor	400.000	510.000	615.000	
Total Operational Costs	400.000	510.000	615.000	
Total Cost	1.070.000	1.020.000	1.230.000	
PV of Costs	345.794,39	102.803,74	107.476,63	556.074,76
PV of All Costs	345.794,39	448.598,13	556.074,76	
Total Project Benefits-Costs	(114.500)	145.500	140.500	
Yearly NPV	(107.009,344)	28.971,962	160.280,378	82.242,996
Cumulative NPV	(107.009,344)	(78.037,382)	82.242,996	
Return on Investment	14.78%			
Break-Even Point	2.48 years (break-even occurs in year 3)			
Intangible Benefits	Increase Image, Accelerate Process, Improved Productivity			

**Gambar. 4 Kelayakan Finansial**

## 4.2 Fase Elaboration

### 4.2.1 Tujuan Aktivitas Proses Elaboration

Elaboration adalah fase kedua dalam siklus pengembangan software. Target fase ini adalah menentukan arsitektur basis sistem yang menjadi landasan disain dan implementasi di fase construction. Target global ini terbagi ke dalam empat tujuan, masing-masing menangani sebuah resiko utama, sebagai berikut: a) Pemahaman kebutuhan yang lebih detail. Selama elaborasi, kita mengupayakan pemahaman yang lebih baik kebutuhan utama, karena kebanyakan baru diuraikan secara singkat dalam inception. Ini memungkinkan kita membuat rencana yang lebih detail; b) Desain, implementasi, validasi dan tentukan arsitektur dasar. Kita perlu untuk mendisain, mengimplementasi, dan menguji struktur kerangka dari sistem. Fungsionalitas di tingkat aplikasi belum akan lengkap, tapi sejalan dengan pengembangan sebagian besar interface antara building block selama fase elaborasi, kita dapat mengkompilasi dan menguji arsitektur sistem. Ini disebut dengan "executable architecture" dimana kita bisa melakukan pengujian awal beban dan kinerja sistem. Di sini kita melakukan keputusan critical mencakup pemilihan teknologi, komponen-komponen utama dan interface-nya; c) Menurunkan resiko utama dan menghasilkan estimasi jadwal dan ongkos lebih akurat. Selama elaborasi, kita mengatasi resiko utama. Dalam prakteknya, resiko ini turun dengan dihasilkannya requirement yang lebih detail disertai disain, implementasi dan pengujian arsitektur. Kita juga memperhalus dan memperdetail rencana kasar proyek; d) Memperhalus pengembangan dan menentukan lingkungan pengembangan. Kita perlu memperhalus proses yang dijalankan dalam inception untuk menggambarkan lesson

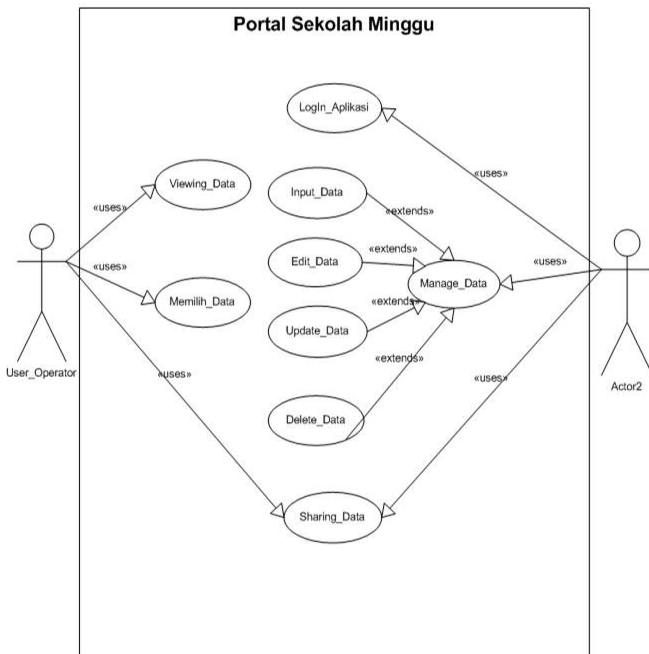
learned. Kita juga melanjutkan implementasi tools pengembangan software yang diperlukan untuk proyek.

**4.2.2 Artifak LCO Fase Elaboration**

Artifak Lifecycle Objectives yang terutama terkait dengan tujuan fase elaboration ini dirangkum dalam dokumen SRS dan dokumen SAD. Berikut adalah artifak terkait tujuan LCO fase elaboration.

a. Functional View (Use Case)

Untuk functional view ditampilkan dengan Use Case Diagram dan Use Case Diagram. Setiap use case dilengkapi dengan use case description. (lihat Gambar 5 dan Gambar 6)



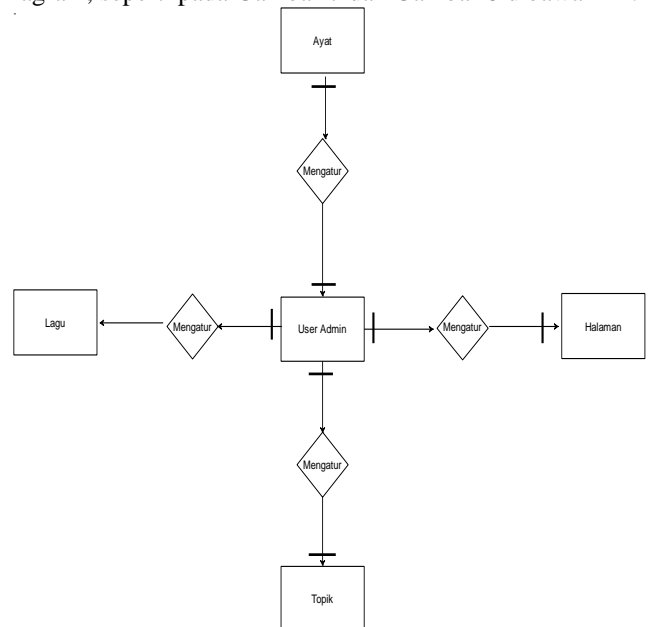
**Gambar. 5 Use Case Diagram**

<b>Use Case Name :</b>	Melihat Menu	
<b>Actor :</b>	Pengguna	
<b>Description :</b>	Memilih pilihan menu yang tersedia sesuai kebutuhan user	
<b>Normal Course :</b>	<b>Actor.Action</b>	<b>System Response</b>
	1. User menekan tombol menu beranda	1. Menampilkan halaman beranda
	2. User menekan tombol pertanyaan	2. Menampilkan halaman pertanyaan
<b>Alternate Course :</b>	<b>Actor.Action</b>	<b>System Response</b>
	1. User menekan tombol keluar	1. Sistem menutup aplikasi
	<b>Precondition :</b> Masuk dalam aplikasi	
<b>Post Condition :</b> Memilih pertanyaan		
<b>Assumption :</b> -		

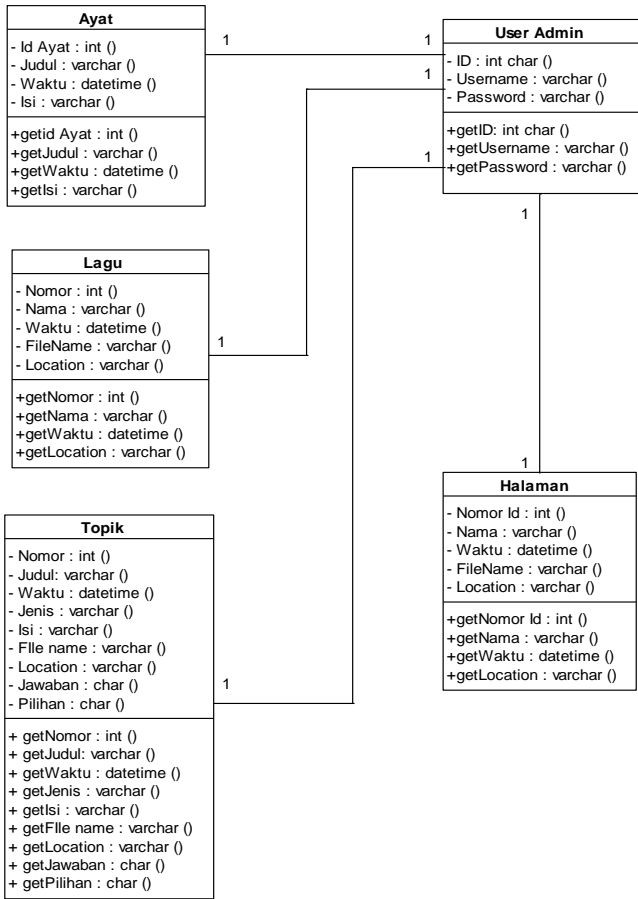
**Gambar. 6 Use Case Description Melihat Menu**

b. Logical View

Logical view dari aplikasi dimodelkan dengan ERD dan Class Diagram, seperti pada Gambar 7 dan Gambar 8 dibawah ini:



**Gambar 7. Entity Relationship Diagram**

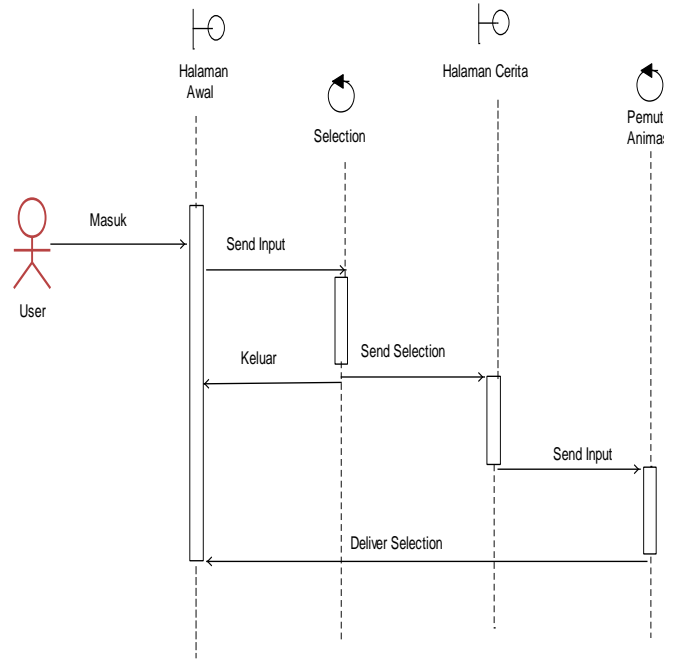


Gambar 8 Gambar Class Diagram

c. Process View

Untuk desain perilaku aplikasi ditampilkan dengan menggunakan sequence diagram dan navigation diagram. Pada Gambar 9 dan Gambar 10 dapat dilihat sequence diagram untuk user dan admin.

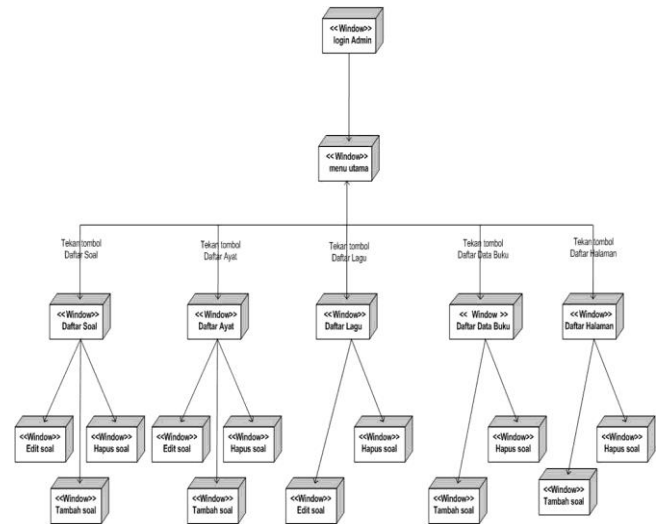
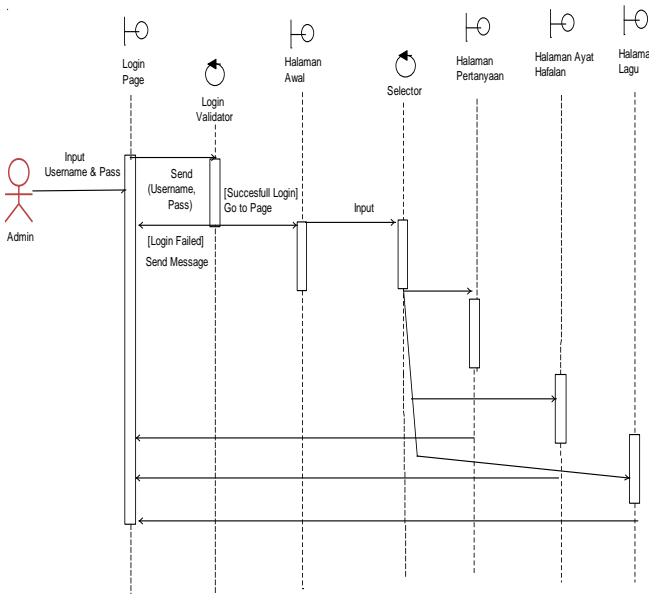
Gambar 9 Sequence Diagram untuk Admin



Gambar. 10 Sequence Diagram untuk User

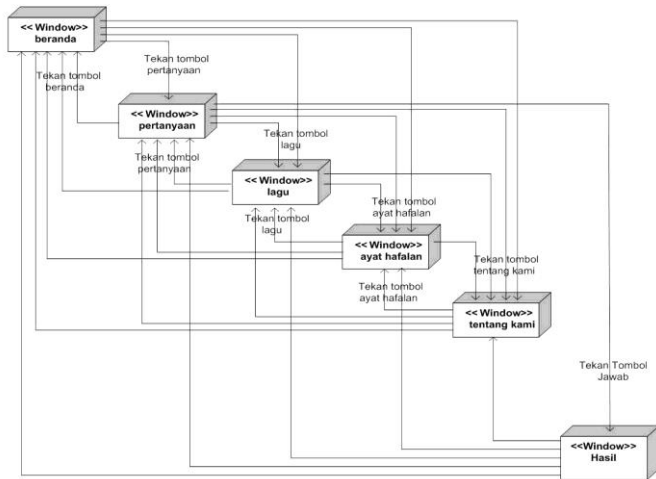
d. Implementation View

Untuk view implementasi digambarkan dengan Navigation Diagram. Gambar 11 menunjukkan model implementasi navigasi dari Admin, sedangkan Gambar. 12 menunjukkan implementasi navigasi untuk User.



Gambar 11 Navigasi Admin





Gambar. 12 Navigasi User

e. Desain Antar-muka

Beberapa screen-shot tampilan disain antar-muka adalah sebagai berikut (lihat Gambar 13 dan Gambar 14):



Gambar. 13 Beranda untuk Operator



Gambar 14 Beranda untuk Admin

4.3 Fase Construction

4.3.1 Tujuan Aktivitas Proses Construction

Target utama fase construction adalah pengembangan yang efisien dan murah menuju produk akhir yaitu versi operasional sistem yang dapat dideploy ke komunitas end-user. Fase ini memiliki tujuan sbb: a) Meminimalisir ongkos pengembangan dan mencapai derajat paralelisme dalam pekerjaan yang dilakukan secara tim.; b) Mengembangkan produk lengkap secara iteratif yang akhirnya siap dipindahkan ke end-use community. Kita mengembangkan versi operasional pertama sistem dengan menggambarkan use case dan requirement yang tersisa, mengisi detail disain, melengkapi implementasi dan menguji software. Kita juga menentukan apakah software, sites dan users semuanya siap untuk aplikasi yang akan dideploy.

4.3.2 Artifak LCO Fase Construction

Artifak LCO yang terutama adalah Source Code aplikasi yang berjalan. Dalam penulisan paper ini hanya ditampilkan script aplikasi tertentu seperti pada Gambar. 15 dibawah.



Gambar. 15 Fitur Login Admin - Tampilan Aplikasi Jadi

Tabel 5 Script Login

Kode Program
<pre> &lt;?php session_start();  if(!isset(\$_SESSION['username'])          &amp;&amp; (!isset(\$_SESSION['password']))) {     echo "&lt;div align=center&gt;&lt;b&gt; PERHATIAN  ...!! &lt;/b&gt;&lt;br&gt;";     echo "KAMU HARUS LOGIN DULU :p &lt;meta http- equiv='Refresh' content='2; Url=index.php/'&gt;&lt;/div&gt;";     exit; } ?&gt; </pre>
<pre> &lt;?php include "koneksi.php"; session_start(); \$username =     trim(strip_tags(mysql_real_escape_string(\$_POST['usern ame']))); \$password =     trim(strip_tags(mysql_real_escape_string(\$_POST['pass </pre>

Kode Program
<pre>word'))); \$password = trim(strip_tags(mysql_real_escape_string(\$_POST['password'])));  if(strlen(\$username)&lt;1) { print "&lt;script&gt;alert('Username Masih Kosong'); javascript:history.go(-1);&lt;/script&gt;"; exit; } else if(strlen(\$password)&lt;1) { print "&lt;script&gt;alert('Password Masih Kosong!'); javascript:history.go(-1);&lt;/script&gt;"; exit; } else if(strlen(\$password2)&lt;1) { print "&lt;script&gt;alert('Password2 Masih Kosong!'); javascript:history.go(-1);&lt;/script&gt;"; exit; } else { \$cari_dulu = "select username, password from useradmin where username = '\$username' and password = '\$password'"; \$jalanin_sql = mysql_query(\$cari_dulu); \$cek_ada_gak = mysql_num_rows(\$jalanin_sql); if (\$cek_ada_gak == 1 and \$password2 == 'admin2') {  \$_SESSION['username'] = \$username; \$_SESSION['password'] = \$password;  header("location: adminarea=3ert"); } else header("location: index.php"); } ?&gt;</pre>
<pre>&lt;?php session_start(); if (\$_REQUEST['logout'] == "yes") session_unset(); session_destroy(); header("location: index.php"); ?&gt;</pre>
<pre>&lt;?php \$my['host'] = "localhost"; \$my['user'] = "root"; \$my['pass'] = ""; \$my['dbs'] = "sad";  \$koneksi = mysql_connect(\$my['host'], \$my['user'], \$my['pass']); if (!\$koneksi) { echo "Gagal Koneksi Bro...."; mysql_error(); }  mysql_select_db(\$my['dbs'] or die ("Database Gak Ada".mysql_error()); ?&gt;</pre>

## 5. KESIMPULAN

- 1) Portal Sekolah Minggu dapat menjadi pilihan solusi untuk kebutuhan Guru dan Anak Sekolah Minggu dalam

mendapatkan kemudahan bahan ajar digital yang interaktif, menarik dan relatif murah.

- 2) Pendekatan agile, dengan kerangka kerja Agile Unified Process (AUP) menjamin pengembangan perangkat lunak yang cepat, namun dilengkapi dengan dokumentasi memadai..

## 6. DAFTAR PUSTAKA

- [1] Schach, *Object Oriented Software Engineering*, 8<sup>th</sup> Ed, McGrawHill, 2008.
- [2] Sommerville, *Software Engineering*, 8<sup>th</sup> ed, Pearson Education Limited, 2007
- [3] Pressman, *Software Engineering, A Practitioner's Approach*, 6<sup>th</sup> ed, McGrawHill, Singapura, 2005.
- [4] Dennis, Wixom and Tegarden, *Sistem Analysis and Design with UML, An Object-Oriented Approach*, 3<sup>rd</sup> ed, John Wiley & Sons, International Student Edition, 2010.
- [5] Bentley and Whitten, *Sistem Analysis and Design for the Global Enterprise*, 7<sup>th</sup> ed, McGrawHill International Edition, 2007.
- [6] Kendall and Kendall, *Sistem Analysis and Design*, 7<sup>th</sup> ed, Pearson Prentice Hall, 2007.
- [7] CMMI Product Team, *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*, November 2010, TECHNICAL REPORT CMU/SEI-2010-TR-033 , ESC-TR-2010-033, Software Engineering Process Management Program, Unlimited distribution subject to the copyright. <http://www.sei.cmu.edu>.
- [8] Martin Fowler, *UML Distilled, A Brief Guide to the Standard Object Modelling Language*, 3<sup>rd</sup> ed, Pearson Education, 2004.
- [9] [www.uml.org](http://www.uml.org)., *Unified Modelling Language: Superstructure Version 2.0*, ptc/03-08-02.
- [10] Philippe Kruchten, *The Rational Unified Process An Introduction*, 3<sup>rd</sup> ed, Pearson Education, 2004.
- [11] Portal Resmi Pusilkom Agile Unified Process (PAUS); <http://ecl.cs.ui.ac.id/PAUS/>
- [12] Portal resmi Ambyssoft <http://www.ambyssoft.com/>