



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI ALBUM FOTO DIGITAL `FOTOKITA` DENGAN PENYIMPANAN DINAMIS

**FADRIAN MERDIANTO
NRP 5112100059**

**Dosen Pembimbing I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.**

**Dosen Pembimbing II
Rizky Januar Akbar, S.Kom., M.Eng.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI ALBUM FOTO DIGITAL `FOTOKITA` DENGAN PENYIMPANAN DINAMIS

**FADRIAN MERDIANTO
NRP 5112100059**

**Dosen Pembimbing I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.**

**Dosen Pembimbing II
Rizky Januar Akbar, S.Kom., M.Eng.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - KI141502

DESIGN AND IMPLEMENTATION OF DIGITAL PHOTO ALBUM `FOTOKITA` USING DYNAMIC STORAGE

**FADRIAN MERDIANTO
NRP 5112100059**

**Supervisor I
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.**

**Supervisor II
Rizky Januar Akbar, S.Kom., M.Eng.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI ALBUM FOTO DIGITAL 'FOTOKITA' DENGAN PENYIMPANAN DINAMIS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

FADRIAN MERDIANTO

NRP : 5112 100 059

Disetujui oleh Dosen Pembimbing

1. Dr.tech. Ir. R.V.HARI GINAR, DTM.Sc.....
NIP: 19650518199203 1 003 (Pembimbing 1)
2. Rizky Januar Akbar, S. Kom., M.Eng.....
NIP: 19871013 201404 1 001 (Pembimbing 2)



SURABAYA

Juni, 2016

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI ALBUM FOTO DIGITAL 'FOTOKITA' DENGAN PENYIMPANAN DINAMIS

Nama Mahasiswa : FADRIAN MERDIANTO
NRP : 5112100059
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R.V. Hari Ginardi,
M.Sc.
Dosen Pembimbing 2 : Rizky Januar Akbar, S. Kom.,
M.Eng.

Abstrak

Semakin maraknya hobi untuk mendokumentasikan kegiatan dalam bentuk foto digital membuat setiap orang semakin tertarik untuk mengunggah foto hasil dokumentasinya ke dalam media-media tertentu. Fenomena tersebut akan menjadi lebih menarik dengan menyatukan dokumentasi tersebut ke dalam sebuah album, yang tentunya juga dalam bentuk digital.

Aplikasi album digital dalam bentuk desktop dapat ditemukan pada aplikasi Fotokita. Dengan kondisi semakin maraknya kolektor foto, tentunya pengguna aplikasi Fotokita berkembang pesat. Dengan penggunaan aplikasi yang intensif tentu diperlukan penyimpanan yang dapat mengakomodasi bermacam gambar serta mampu mengakomodasi berbagai versi agar tidak ada kesalahan ketika membuka file. Pada tugas akhir ini dibuat aplikasi yang mampu mengakomodasi penyimpanan serta pembacaan file berdasarkan informasi metadata pada aplikasi Fotokita sesuai dengan hasil yang sama dengan aplikasi. Rancangan ini dapat mengakomodasi perubahan struktur file untuk mengantisipasi pembaruan aplikasi dengan konsep forward compatibility dan backward compatibility.

Kata kunci: Foto Digital, meta-data, forward compatibility, backward compatibility

[Halaman ini sengaja dikosongkan]

DESIGN AND IMPLEMENTATION OF DIGITAL PHOTO ALBUM 'FOTOKITA' USING DYNAMIC STORAGE

Student's Name : FADRIAN MERDIANTO
Student's ID : 5112100059
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.tech. Ir. R.V. Hari Ginardi,
M.Sc.
Second Advisor : Rizky Januar Akbar, S. Kom.,
M.Eng.

Abstract

Increasingly widespread hobby to document activities in the form of digital photos causes everyone becomes more interested to upload photos of documentation results in certain media. The phenomenon will become more attractive to unify the documentation into an album, which of course also in digital form.

Desktop-based digital album can be found on Fotokita application. With the increasingly widespread condition of photos collector, Fotokita's user will be growing rapidly. With the use of intensive applications, storage that can accommodate a variety of images and able to accommodate the various versions so no error will occur when opening the file is certainly needed. In this final project application that is able to accommodate the storage and reading of the file Fotokita application in accordance with the same results with the application is built. This design can accommodate changes in the file structure to anticipate different structure in shape of forward and backward compatibility, with certain provisions.

Keyword: Digital Photo Album, meta-data, forward compatibility, backward compatibility

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

“Rancang Bangun Aplikasi Album Foto Digital ‘Fotokita’ dengan Penyimpanan Dinamis”.

Tugas akhir ini dilakukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama proses pengerjaan tugas akhir ini hingga selesai, antara lain:

1. Allah SWT dan Nabi Muhammad SAW.
2. Kedua orang tua, yang selalu memberikan semangat dan perhatian sejak saya menjadi mahasiswa baru di tahun 2012 hingga dapat menyelesaikan tugas akhir ini.
3. Kakak serat adik, yang menjadi salah satu semangat agar dapat menyelesaikan tugas akhir ini dan melanjutkan ke tingkat yang lebih jauh.
4. Bapak Hari Ginardi, selaku dosen pembimbing 1 yang telah memberikan bimbingan dan masukan mengenai tugas akhir ini.
5. Bapak Rizky Januar Akbar, selaku dosen pembimbing 2 yang telah meluangkan waktu dan pikiran untuk berdiskusi mengenai teknis pengerjaan tugas akhir.
6. Teman-teman seperjuangan, Rina, Dwi, Dala, dan Ridwan, yang selama satu semester lebih berjuang bersama-sama menyelesaikan aplikasi Fotokita.
7. Teman-teman Teknik Informatika ITS angkatan 2012, kakak maupun adik di kampus, yang dengan sangat senang hati

membantu, berbagi ilmu di Teknik Informatika ITS selama masa kuliah saya.

8. Bapak dan Ibu dosen pengajar, yang telah berbagi ilmu dan mendidik saya dan seluruh mahasiswa Informatika ITS selama kuliah.
9. Teman-teman Kabinet Berkarya HMTC FTif ITS 2014-2015, serta seluruh teman-teman HMTC yang telah menjadi keluarga ketika jauh dari kampung halaman.
10. Ridlo Iqbal, Darma, Agil, dan Aim yang telah dan akan terus menjadi saudara kedua.
11. Teman-teman lain yang tidak dapat disebutkan namanya satu persatu, yang terus memberikan semangat dan hiburan serta dukungan dari luar ketika sedang dalam keadaan baik ataupun kurang baik.
12. Stackoverflow, yang seringkali memberikan jawaban atas pertanyaan yang sering muncul mengenai masalah pemrograman yang saya temukan.

Saya mohon maaf apabila terdapat kekurangan dalam penulisan buku tugas akhir ini. Kritik dan saran saya harapkan untuk perbaikan dan pembelajaran di kemudian hari. Semoga tugas akhir ini dapat memberikan manfaat yang sebaik-baiknya.

Surabaya, Juni 2016

Fadrian Merdianto

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Metodologi.....	4
1.7 Sistematika Penulisan Laporan Tugas Akhir	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 XML.....	7
2.2 XPath.....	7
2.3 Java JAXB 2.0.....	9
2.4 Java DOM.....	10
2.5 Skema XML.....	10
2.6 Forward Compatibility	10
2.7 Backward Compatibility.....	11
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	13
3.1 Analisis	13
3.1.1 Analisis Desain Utama Aplikasi.....	14
3.1.2 Analisis Desain Skema Validasi.....	15
3.1.3 Analisis Desain Forward Compatibility dan Backward Compatibility.....	16
3.2 Perancangan	16
3.2.1 Perancangan Struktur File.....	16
3.2.2 Perancangan Diagram Kelas XML.....	17

3.2.3	Perancangan Skema Validasi.....	24
3.2.4	Perancangan Skema Forward dan Backward Compatibility	30
3.2.5	Perancangan <i>Partial Update</i>	32
BAB IV	IMPLEMENTASI.....	33
4.1	Lingkungan Implementasi	33
4.2	Implementasi Struktur File	33
4.3	Penggunaan Masing-masing Kelas.....	35
4.3.1	Penggunaan Kelas Pada XML.....	35
4.3.2	Penggunaan Kelas pada Skema Validasi XML	38
4.4	Implementasi Fungsi Tulis dan Baca XML menggunakan JAXB dan DOM.....	39
4.4.1	Fungsi Marshalling JAXB.....	39
4.4.2	Fungsi Unmarshalling menggunakan DOM.....	40
4.4.3	Penggunaan Fungsi cekSchema().....	41
4.5	Penggunaan Library dan Template	42
4.5.1	Penggunaan Library UUID.....	42
4.5.2	Penggunaan Library JAXB.....	42
4.6	Implementasi Forward dan Backward Compatibility	45
4.6.1	Implementasi Forward Compatibility	45
4.6.2	Implementasi Backward Compatibility.....	45
BAB V	PENGUJIAN DAN EVALUASI	47
5.1	Lingkungan Pengujian.....	47
5.2	Skenario Pengujian Fungsionalitas.....	47
5.2.1	Pengujian Menyimpan File Proyek	48
5.2.2	Pengujian Membaca File Proyek dengan Versi Aplikasi yang sama.....	50
5.2.3	Pengujian Menyimpan File Proyek yang Telah diubah.....	52
5.2.4	Pengujian Membaca File Proyek dengan Versi Aplikasi yang berbeda.....	56
5.3	Skenario Pengujian Konsistensi	61
BAB VI	KESIMPULAN DAN SARAN.....	63
6.1	Kesimpulan	63
6.2	Saran.....	63

DAFTAR PUSTAKA	65
BIODATA PENULIS	67

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1.1 Diagram Arsitektur Sistem Aplikasi Fotokita.....	2
Gambar 1.2 <i>Grand Design</i> Aplikasi	3
Gambar 2.1 Arsitektur JAXB.....	9
Gambar 2.2 Skema kerja forward compatibility dan backward compatibility	11
Gambar 3.1 Diagram Kasus Penggunaan Aplikasi Fotokita.....	13
Gambar 3.2 Diagram Pohon Susunan XML	15
Gambar 3.3 Struktur file .fbk.....	17
Gambar 3.4 Diagram Kelas Model Data XML	18
Gambar 3.5 Kelas Fotokita	19
Gambar 3.6 Kelas XMLPages.....	20
Gambar 3.7 Kelas XMLPage.....	20
Gambar 3.8 Kelas XMLHolder.....	21
Gambar 3.9 Kelas XMLFrameholder	22
Gambar 3.10 Kelas XMLPhotoholder	23
Gambar 3.11 Kelas Paragraph.....	24
Gambar 3.12 Kelas Diagram Skema Validasi XML.....	25
Gambar 3.13 Kelas Fotokita	26
Gambar 3.14 Kelas Pages.....	27
Gambar 3.15 Kelas Page	27
Gambar 3.16 Kelas Holder	28
Gambar 3.17 Kelas Frameholder.....	29
Gambar 3.18 Kelas Photoholder.....	29
Gambar 3.19 Kelas Paragraph.....	30
Gambar 3.20 Perancangan Skema Forward Compatibility.....	31
Gambar 3.21 Perancangan Skema Backward Compatibility	32
Gambar 4.1 Struktur objek penyusun file .fbk.....	34
Gambar 4.2 Struktur objek elemen pada file .fbk.....	34
Gambar 5.1 Hasil Proyek yang Telah Dibuat	49
Gambar 5.2 Dialog Penyimpanan File Proyek.....	49
Gambar 5.3 File Hasil Penyimpanan	50
Gambar 5.4 Dialog Membuka File Proyek.....	51
Gambar 5.5 Tampilan Hasil File Proyek yang Telah dibaca	52

Gambar 5.6 Hasil Proyek yang Telah Dibuka	54
Gambar 5.7 Proyek Setelah Diubah.....	55
Gambar 5.8 Susunan Metadata Ketika dibaca	54
Gambar 5.9 Susunan Metadata setelah Proyek diubah	55
Gambar 5.10 Pemberitahuan Untuk Memperbarui Aplikasi	57
Gambar 5.11 Struktur Metadata File Proyek Versi Baru.....	58
Gambar 5.12 File Versi Baru yang dibuka pada Aplikasi Versi Lebih Rendah.....	58
Gambar 5.13 Struktur Metadata File Proyek Versi Lebih Rendah	59
Gambar 5.14 File Versi Lebih Rendah yang dibuka pada Aplikasi Versi Lebih Tinggi.....	60

DAFTAR TABEL

Tabel 2.1 Standard XPath.....	8
Tabel 3.1 Objek penyusun <i>metadata</i>	14
Tabel 4.1 Implementasi JAXB pada Dokumen XML.....	43
Tabel 5.1 Spesifikasi Pengujian.....	47
Tabel 5.2 Skenario Uji Coba Menyimpan File Proyek.....	48
Tabel 5.3 Skenario Uji Coba Membaca File Proyek.....	51
Tabel 5.4 Skenario Uji Coba Menyimpan File Proyek yang telah diubah.....	52
Tabel 5.5 Skenario Uji Coba Skema Forward Compatibility.....	56
Tabel 5.6 Skenario Uji Coba Backward Compatibility.....	60
Tabel 5.7 Hasil Pengujian Konsistensi.....	61

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Fungsi Kelas Validasi.....	37
Kode Sumber 4.2 Implementasi Fungsi Tulis (Marshalling).....	40
Kode Sumber 4.3 Penggunaan Fungsi Unmarshalling.....	41
Kode Sumber 4.4 Penggunaan Fungsi cekSchema.....	41
Kode Sumber 4.5 Penggunaan Library UUID.....	42
Kode Sumber 4.7 Implementasi JAXB pada Atribut “version” ..	43
Kode Sumber 4.8 Implementasi JAXB pada Elemen “title”	43
Kode Sumber 4.9 Implementasi JAXB pada Value.....	43

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

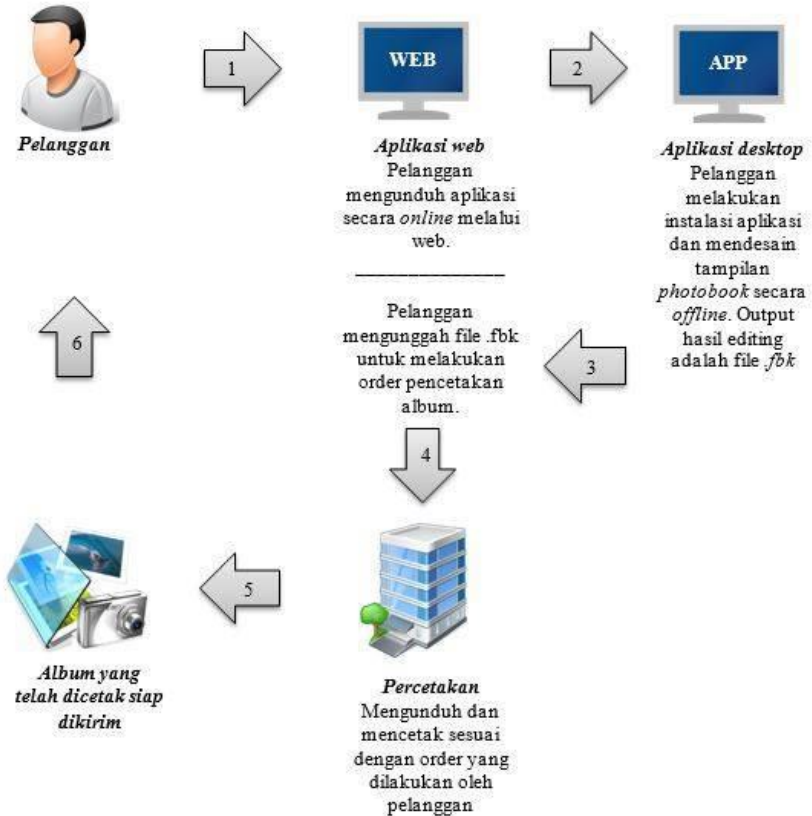
Pada bab ini akan dijelaskan hal-hal yang menjadi latar belakang, permasalahan yang dihadapi, batasan masalah, tujuan, metodologi dan sistematika penulisan yang digunakan dalam pembuatan buku tugas akhir ini.

1.1 Latar Belakang

Fenomena semakin maraknya aktivitas pengabdian momen, dan diiringi dengan kebutuhan masyarakat, tentunya harus ada teknologi yang dapat membantu memudahkan proses tersebut. Beberapa solusi yang pernah ada belum maksimal, dikarenakan mayoritas solusi tersebut ada dalam bentuk *online*, sedangkan jaringan di Indonesia belum bisa dikatakan mampu untuk melakukan semua kegiatan dalam sistem *online*. Perlu adanya sistem yang memudahkan masyarakat untuk berkreasi menyusun hasil jepretan mereka secara *standalone*, agar pengerjaan tersebut berjalan stabil, dan tentunya perlu adanya tempat untuk memamerkan hasil karya masyarakat secara luas, sehingga perlu adanya sistem yang juga mendukung pertunjukan karya tersebut secara *online*.

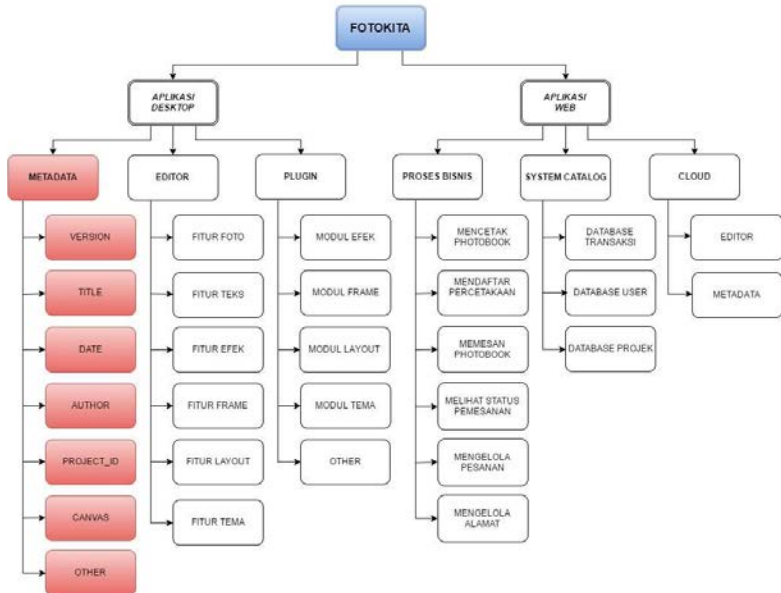
Dengan berbagai struktur dalam buku tersebut, dan dengan kebutuhan bahwa dalam aplikasi tersebut harus dapat dengan baik tersinkronisasi ketika *offline* maupun *online*, tentunya dibutuhkan sebuah cara agar berbagai jenis kreasi pengguna tersebut dapat disimpan dalam sebuah *file* yang dapat digunakan kembali. Dalam penyimpanan *file* tersebut digunakan XML, sebuah bahasa *markup* untuk membuat dokumen *markup* keperluan pertukaran antara sistem yang berbeda. Dengan XML, banyak komponen yang dibutuhkan untuk membuat struktur yang baik agar dalam penyimpanannya, struktur-struktur *file* tersebut dapat disusun secara struktural.

Secara garis besar, jalannya alur penggunaan sistem aplikasi Fotokita dapat dijabarkan dalam diagram alur Gambar 1.1.



Gambar 1.1 Diagram Arsitektur Sistem Aplikasi Fotokita

Dari diagram arsitektur sistem aplikasi Fotokita tersebut, bagian yang diimplementasi dalam tugas akhir ini adalah proses manajemen informasi *metadata* agar informasi yang diterima dan dibuat oleh aplikasi dapat berjalan tanpa hambatan. *Grand design* aplikasi Fotokita dapat dilihat pada Gambar 1.2. Bagan berwarna biru merupakan desain aplikasi secara keseluruhan, bagian berwarna merah merupakan bagian sistem yang dikerjakan pada tugas akhir ini.



Gambar 1.2 Grand Design Aplikasi

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat desain struktur *metadata* dalam aplikasi fotokita yang dapat mengakomodasi perubahan?
2. Bagaimana agar *file* yang disimpan dapat digunakan dalam aplikasi yang dapat memenuhi aspek *backward compatibility* dan *forward compatibility*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi berbasis *desktop* dengan bahasa pemrograman JAVA.
2. Aplikasi dapat disimpan ke dalam *file* bentuk ekstensi *.fbk*.
3. *File* dapat disimpan dan dibuka dengan aplikasi berbeda versi.

1.4 Tujuan

Tugas akhir ini mempunyai beberapa tujuan, yaitu sebagai berikut:

1. Menyusun desain struktur *metadata* aplikasi 'Fotokita'.
2. Menentukan metode penyimpanan yang dapat dibuka dengan berbagai versi.
3. Meningkatkan efisiensi penyimpanan *file* dengan *partial updating*.

1.5 Manfaat

Manfaat yang diharapkan dari pengembangan aplikasi ini adalah untuk memudahkan pengguna mengatur dan mencetak album foto sesuai dengan kreasi masing-masing, dalam hal ini pada proses penyimpanan.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penyusunan proposal tugas akhir.
Tahap awal untuk memulai pengerjaan tugas akhir adalah penyusunan proposal tugas akhir. Proposal tugas akhir yang diajukan memiliki gagasan yang sama dengan tugas akhir ini, yaitu rancang bangun aplikasi editor album foto digital 'Fotokita'.
2. Studi literatur
Pada tahap ini dilakukan pengamatan terhadap beberapa aplikasi yang telah ada di pasaran. Pengamatan dilakukan terhadap aplikasi Photobook Indonesia serta beberapa aplikasi dari luar lainnya. Studi literatur juga melalui buku album foto dalam bentuk fisik yang merupakan hasil dari pencetakan album dalam bentuk digital.
3. Analisis dan desain perangkat lunak
Tahap ini meliputi perumusan kebutuhan kelas diagram, rancangan alur penyimpanan *metadata* untuk pembuatan penyimpanan tugas akhir.

4. Implementasi perangkat lunak
Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan rancangan yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.
5. Pengujian dan evaluasi
Pada tahapan ini dilakukan uji coba pada perangkat lunak dengan pengujian melalui prototipe dan implementasi pada aplikasi album foto digital `Fotokita`.
6. Penyusunan buku tugas akhir
Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang desain sistem, rancangan basis data, diagram kasus penggunaan, diagram aktivitas dan rancangan antarmuka pengguna.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan rancang bangun penyimpanan sistem aplikasi album foto digital. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 XML

XML (*eXtensible Markup Language*) adalah sebuah bahasa markah untuk mendeskripsikan data. XML merupakan turunan (subset) atau versi ringkas dari SGML (*Standard Generalized Markup Language*). SGML sendiri merupakan sebuah standar ISO untuk format dokumen. SGML tidak berisi berupa tag-tag siap pakai seperti halnya bahasa HTML, melainkan berupa aturan-aturan standar dalam pembuatan tag-tag format dokumen. SGML banyak dipakai untuk mengelola dokumen dalam jumlah besar, frekuensi revisi tinggi dan dibutuhkan dalam beragam format tampilan. SGML jarang dipakai karena sangat rumit dan kompleks. XML dibuat dengan konsep yang lebih sederhana dan ringkas, tujuannya agar bisa dipakai sebagai aplikasi di *desktop* dan jaringan Internet [1].

XML dengan cepat diadopsi sebagai standar untuk pertukaran data, khususnya untuk penggunaan lintas aplikasi dan *platform*. Kelebihan dari XML adalah karakteristiknya yang *extensible* dan *platform* independen. Berbeda dengan HTML, XML tidak mempunyai kosakata (berupa "tag") yang baku, sebaliknya dengan XML kita bebas merancang tag-tag sendiri, sesuai dengan kebutuhan aplikasinya.

2.2 XPath

XPath (XML Path Language) adalah bahasa *query* untuk memilih bagian - bagian (*node*) dari sebuah dokumen XML. XPath juga dapat digunakan untuk menghitung nilai (contoh: *string*,

angka atau *boolean*) dari isi dari sebuah dokumen XML. XPath distandarisasi oleh *World Wide Web Consortium* (W3C). Bahasa XPath dirancang berdasarkan struktur pohon dari dokumen XML, dan memiliki kemampuan untuk menavigasi struktur pohon tersebut serta memilih bagian-bagian dari dokumen XML berdasarkan berbagai macam kriteria [3].

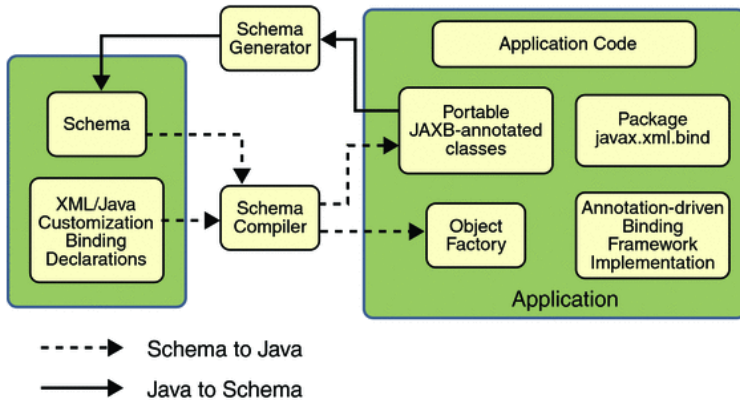
XPath mempunyai standard untuk setiap ekspresi *query* yang dilakukan. Standard XPath tersebut dijabarkan pada Tabel 2.1 [2].

Tabel 2.1 Standard XPath

Ekspresi	Deskripsi
<i>node</i>	Memilih seluruh <i>node</i> dengan nama “ <i>node</i> ”
/	Memilih <i>node</i> berdasarkan <i>root</i>
//	Memilih <i>node</i> dalam dokumen XML yang memiliki nama sesuai ekspresi
.	Memilih <i>node</i> saat ini
..	Memilih <i>parent</i> dari <i>node</i> saat ini
@	Memilih <i>attribute</i> pada <i>node</i>
<i>/element</i>	Memilih <i>element</i> dengan <i>root</i> yang bernama “ <i>element</i> ”
<i>element/child</i>	Memilih <i>element</i> yang merupakan <i>child</i> dari “ <i>element</i> ”
<i>element//child</i>	Memilih <i>element</i> yang merupakan turunan dari <i>root</i> “ <i>element</i> ”, meskipun bukan <i>child</i> dari <i>root</i> “ <i>element</i> ”
[<i>indeks</i>]	Memilih <i>element</i> berdasarkan indeks
[<i>last()</i>]	Memilih <i>element</i> indeks terakhir
[<i>position() < n</i>]	Memilih <i>element</i> indeks 0 sampai <i>n</i>
<i>//element[@attr=“value”]</i>	Memilih <i>element</i> yang memiliki <i>attribute</i> “ <i>attr</i> ” yang bernilai “ <i>value</i> ”
<i>//element[val]</i>	Memilih <i>element</i> memiliki <i>value</i> “ <i>val</i> ”
<i>//element[val1</i> <i>//element[val2</i>	Memilih <i>element</i> “ <i>val1</i> ” dan “ <i>val2</i> ” yang merupakan <i>child</i> “ <i>element</i> ”

2.3 Java JAXB 2.0

Java Architecture for XML Binding (JAXB) menyediakan cara cepat dan nyaman untuk mengikat skema XML dan representasi Jawa, sehingga mudah bagi pengembang Java untuk menggabungkan data XML dan fungsi pengolahan dalam aplikasi Java. Sebagai bagian dari proses ini, JAXB menyediakan metode untuk *unmarshalling* (membaca) XML contoh dokumen ke pohon konten Java, dan kemudian menyusun pohon (menulis) konten Java kembali ke dokumen XML misalnya. JAXB juga menyediakan cara untuk menghasilkan skema XML dari objek Java. Komponen yang menyusun implementasi JAXB dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur JAXB

Keterangan dari Gambar 2.1 adalah sebagai berikut [3].

- **Schema compiler:** Membungkus skema sumber ke dalam sebuah elemen skema turunan program.
- **Schema generator:** Memetakan elemen-elemen yang telah dibuat berdasarkan program ke dalam skema turunan.
- **Binding runtime frameworks:** menyediakan *unmarshalling* (membaca) dan *marshalling* (menulis).

2.4 Java DOM

Document Object Model merupakan rekomendasi resmi dari *World Wide Web Consortium* (W3C). Ini mendefinisikan sebuah antarmuka yang memungkinkan program untuk mengakses dan memperbarui gaya, struktur, dan isi dari dokumen XML. XML parser yang mendukung DOM mengimplementasikan *interface*-nya.

2.5 Skema XML

Skema XML merupakan sebuah bahasa untuk mengekspresikan batasan-batasan yang dapat ditentukan untuk sebuah dokumen XML. Ada banyak jenis bahasa skema XML yang dapat digunakan dan tersebar dapat digunakan, yang paling umum diantaranya adalah *Document Type Definitions* (DTDs), Relax-NG, Schematron, dan W3C XSD (*XML Schema Definition*).

Skema XML dapat digunakan sebagai [2] [4].

- Mendefinisikan daftar elemen yang digunakan.
- Untuk mengasosiasikan tipe data.
- Memberikan batasan atribut dan elemen yang dapat diletakkan dalam suatu *node*.
- Memudahkan manusia untuk membaca dokumen XML.
- Sebagai bentuk normal dari definisi dokumen XML satu dengan yang lainnya.

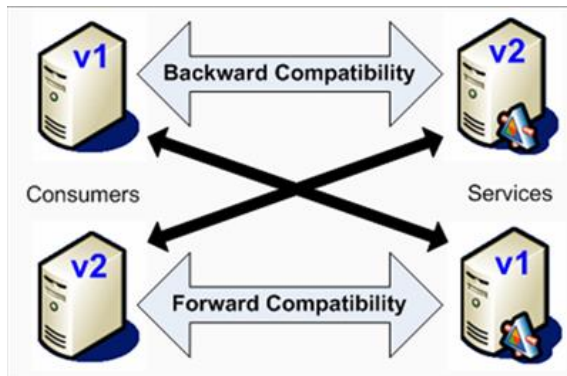
2.6 Forward Compatibility

Kemampuan sebuah sistem untuk menerima, membaca, melihat, ataupun melakukan fitur pada sistem baru yang memiliki komponen berbeda, meskipun tidak seluruh fitur dari sistem yang baru tersebut diimplementasikan. Contoh dari *forward compatibility* adalah aplikasi Microsoft Office Word 2003, yang dapat membuka *file* proyek aplikasi Microsoft Office Word 2007, namun tidak mengimplementasi beberapa fitur seperti sitasi, persamaan secara optimal [5].

2.7 Backward Compatibility

Kemampuan sebuah sistem baru untuk menerima, membaca, melihat, ataupun melakukan fitur pada sistem sebelumnya yang memiliki komponen berbeda. Contoh implementasi *backward compatibility* antara lain adalah perangkat PlayStation 3 dapat memainkan permainan yang ada pada PlayStation 2.

Perbedaan antara *forward compatibility* dan *backward compatibility* dapat dilihat pada gambar Gambar 2.2 [6].



Gambar 2.2 Skema kerja forward compatibility dan backward compatibility

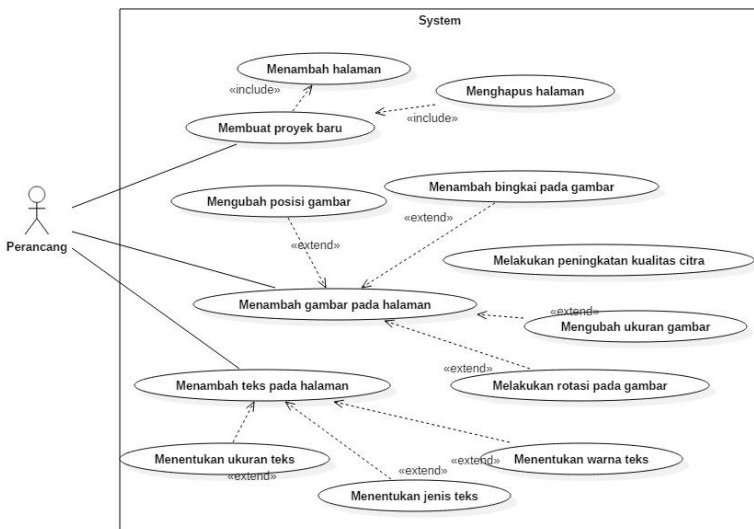
BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini membahas analisis dan perancangan sistem yang dibuat pada tugas akhir ini. Penjelasan tersebut meliputi analisis desain utama sistem, analisis desain skema validasi analisis desain *forward compatibility* dan *backward compatibility*.

3.1 Analisis

Secara umum sistem yang dibuat merupakan sistem penyimpanan dari tugas akhir yang berjudul “Rancang Bangun Aplikasi Editor Album Foto Digital 'Fotokita' Berbasis Desktop” [7]. Kasus penggunaan dapat dilihat pada Gambar 3.1. Penyimpanan dari aplikasi tersebut disimpan menjadi sebuah *file* .fbk yang tersusun dari susunan *metadata*. *File* yang telah dibuat dapat dibuka kembali oleh aplikasi di lain waktu. Selain itu, proyek juga dapat diunggah ke dalam aplikasi katalog *web* untuk nantinya dapat diproses dan dicetak sesuai kebutuhan pengguna.



Gambar 3.1 Diagram Kasus Penggunaan Aplikasi Fotokita

3.1.1 Analisis Desain Utama Aplikasi

Aplikasi terdiri dari dua bagian, yaitu berbasis *desktop* yang digunakan untuk melakukan pengubahan proyek, dan berbasis *web* untuk manajemen katalog proyek. Pada aplikasi utama *desktop*, pengguna dapat melakukan beberapa proses sebagai berikut.

- Membuat proyek baru.
- Menambahkan halaman.
- Menambahkan gambar pada masing-masing halaman.
- Mengatur frame untuk masing-masing gambar.
- Mengubah proyek.

Pada aplikasi utama web, pengguna dapat melakukan beberapa proses sebagai berikut.

- Mengunggah proyek yang telah ada ke aplikasi web.
- Mengatur pencetakan proyek.

Berdasarkan analisis yang telah dijelaskan sebelumnya, disusun objek yang dibutuhkan agar kebutuhan pada aplikasi *desktop* dan *web* dapat diakomodasi. Setiap *file* yang disusun untuk tugas akhir ini memenuhi beberapa objek yang dijabarkan pada Tabel 3.1.

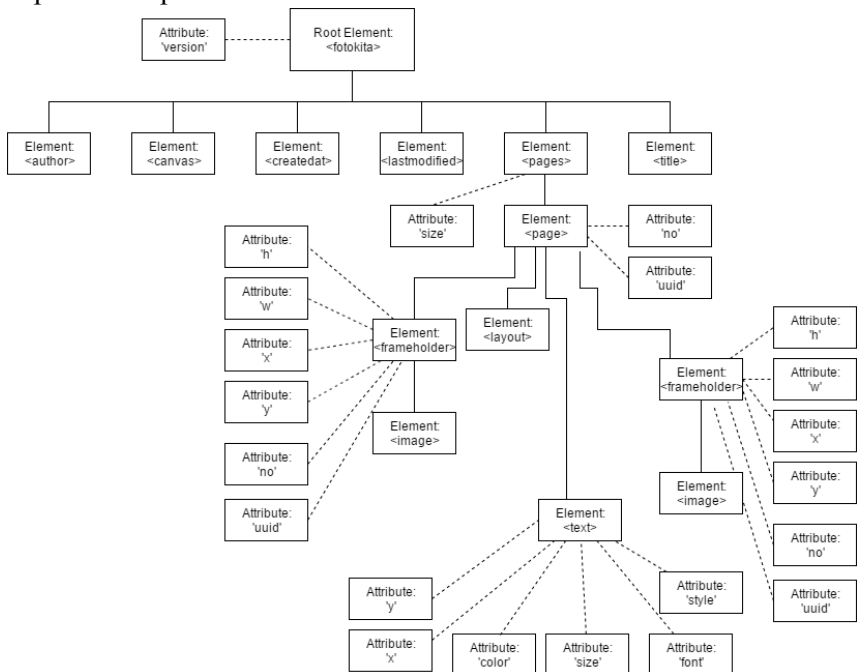
Tabel 3.1 Objek penyusun *metadata*

No	Objek	Keterangan
1.	Judul Proyek	Judul Proyek yang dibuat
2.	Versi Aplikasi	Versi Aplikasi pada saat pembuatan proyek
3.	Pembuat Proyek	Pembuat proyek
4.	Jumlah Halaman	Banyaknya halaman yang dibuat dalam satu proyek
5.	Tanggal Pembuatan Proyek	Tanggal pembuatan pertama proyek
6.	Pengubahan Terakhir Proyek	Tanggal perubahan terakhir proyek
7.	Ukuran Halaman	Ukuran halaman proyek
8.	Gambar Foto	Gambar foto yang disusun
9.	Frame	Frame penampung gambar

No	Objek	Keterangan
10.	Teks	Teks yang ditambahkan untuk menambah keterangan gambar objek

3.1.2 Analisis Desain Skema Validasi

Validasi skema penyimpanan *metadata* dilakukan setiap sesaat sebelum aplikasi menyimpan *file* dan sesaat sebelum aplikasi membuka *file* proyek. Hal ini dilakukan untuk mencegah terjadinya kesalahan dalam pembuatan struktur XML. Validasi skema juga dilakukan untuk mengakomodasi keberadaan situasi yang mengakibatkan *forward compatibility* maupun *backward compatibility*. Susunan analisis struktur XML aplikasi Fotokita dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Pohon Susunan XML

3.1.3 Analisis Desain Forward Compatibility dan Backward Compatibility

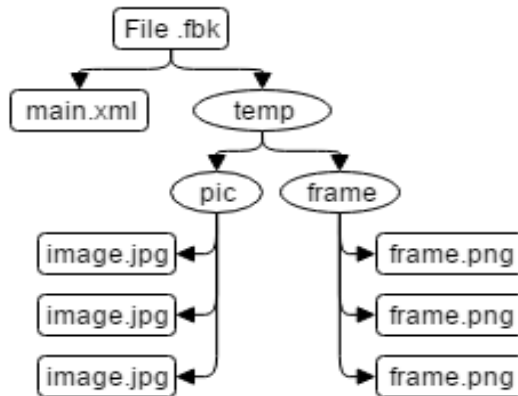
Pada kondisi aplikasi tipe versi lebih rendah membaca *file* berekstensi *.fbk* yang disimpan dengan versi lebih tinggi, maka aplikasi hanya akan mengimplementasi fitur pada *file* berekstensi *.fbk* yang juga terdapat pada aplikasi versi lebih rendah tersebut. Sedangkan pada kondisi aplikasi tipe versi lebih tinggi ketika membaca *file* berekstensi *.fbk* yang disimpan dengan versi lebih rendah, maka aplikasi akan mengimplementasi seluruh fitur yang ada pada *file .fbk* tersebut.

3.2 Perancangan

Pada subbab perancangan akan dijelaskan mengenai perancangan arsitektur sistem yang dibuat dalam tugas akhir ini. Perancangan terdiri dari perancangan diagram kelas XML, perancangan skema validasi, perancangan skema *forward compatibility* dan *backward compatibility*, serta perancangan *partial update*.

3.2.1 Perancangan Struktur File

File dibungkus dalam format ekstensi *.fbk*. Struktur *file* dapat dilihat pada Gambar 3.4. Di dalam bungkus tersebut terdapat dua elemen, yaitu folder *temp* dan *file main.xml*. *File main.xml* merupakan *file* utama yang berisi struktur xml yang digunakan untuk membungkus maupun membuka file dengan ekstensi *.fbk*. Di dalam folder *temp* terdapat folder *pic* dan *frame*, yang menampung gambar yang disimpan dan dibuka oleh aplikasi. Gambar yang disimpan di dalam folder *pic* dan *frame* merupakan file binary yang nantinya akan langsung dimuat dan diimplementasikan pada editor ketika file tersebut dibuka oleh aplikasi.



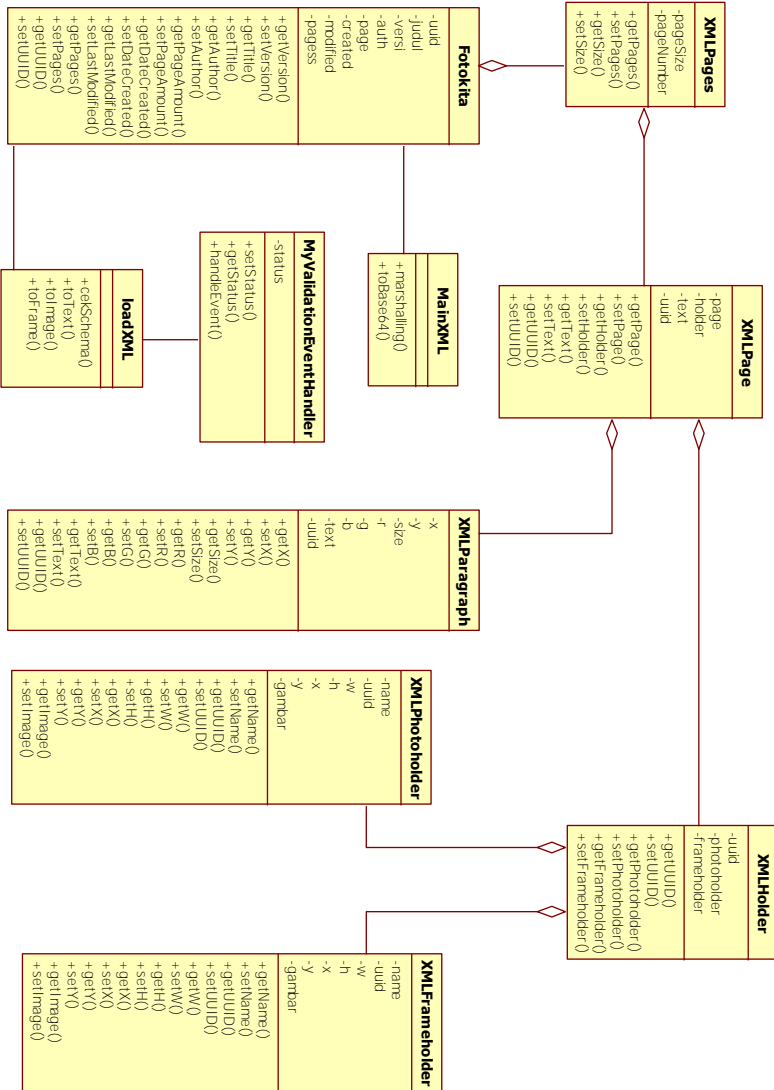
Gambar 3.3 Struktur file .fbk

3.2.2 Perancangan Diagram Kelas XML

Gambar 3.4 menunjukkan diagram kelas dalam sistem tugas akhir ini. Penjelasan masing-masing kelas yang terdapat pada diagram tersebut akan dijelaskan lebih lanjut pada subbab selanjutnya.

3.2.2.1 Kelas Fotokita

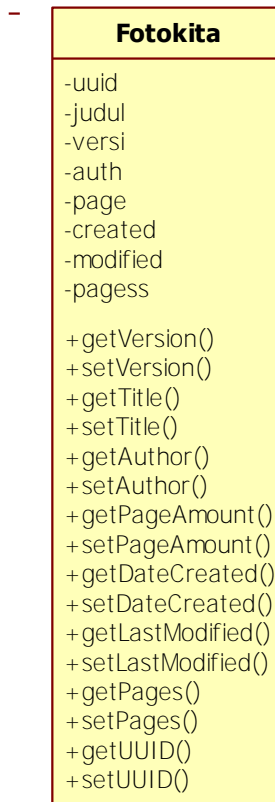
Kelas *Fotokita* menangani setiap perubahan maupun masukan data pada beberapa elemen utama, antara lain versi aplikasi, judul aplikasi, pemilik proyek, tanggal proyek dibuat, serta tanggal terakhir pengubahan proyek. Pada kelas ini terdapat *handler* berupa fungsi *setter* dan *getter* yang menampung informasi dari aplikasi. Nilai yang ada pada elemen *title*, *uuid*, *version*, *author*, *createdat*, serta *lastmodified*. Susunan atribut dan fungsi yang ada pada Kelas *Fotokita* dapat dilihat pada Gambar 3.5.



Gambar 3.4 Diagram Kelas Model Data XML

3.2.2.2 Kelas XMLPages

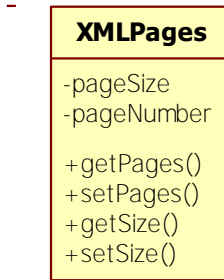
Kelas XMLPages mengatur kelas turunannya yaitu Kelas Page. Kelas ini terdiri dari ukuran halaman serta terdiri dari 0...n objek dari Kelas Page. Objek Page dari Kelas Pages ditampilkan dalam bentuk *list* untuk memudahkan dalam menampung jumlah halaman yang dinamis. Susunan atribut dan fungsi yang ada pada Kelas Pages dapat dilihat pada Gambar 3.6.



The diagram shows a class named 'Fotokita' with a list of attributes and methods. The attributes are listed with a minus sign (-) and the methods with a plus sign (+). The attributes are: -uuid, -judul, -versi, -auth, -page, -created, -modified, and -pagess. The methods are: +getVersion(), +setVersion(), +getTitle(), +setTitle(), +getAuthor(), +setAuthor(), +getPageAmount(), +setPageAmount(), +getDateCreated(), +setDateCreated(), +getLastModified(), +setLastModified(), +getPages(), +setPages(), +getUUID(), and +setUUID().

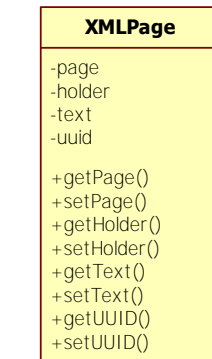
Fotokita
-uuid
-judul
-versi
-auth
-page
-created
-modified
-pagess
+getVersion()
+setVersion()
+getTitle()
+setTitle()
+getAuthor()
+setAuthor()
+getPageAmount()
+setPageAmount()
+getDateCreated()
+setDateCreated()
+getLastModified()
+setLastModified()
+getPages()
+setPages()
+getUUID()
+setUUID()

Gambar 3.5 Kelas Fotokita



Gambar 3.6 Kelas XMLPages

3.2.2.3 Kelas Page

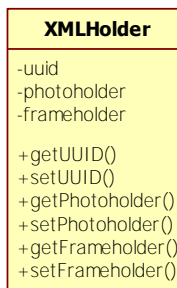


Gambar 3.7 Kelas XMLPage

Kelas `XMLPage` pada Gambar 3.7 merupakan turunan dari Kelas `XMLPages`, di mana kelas ini mempunyai atribut berupa informasi yang dapat dibaca oleh aplikasi serta ditulis oleh aplikasi. Informasi tersebut berupa teks yang ada pada halaman tersebut. Setiap halaman dapat memuat 0..n objek teks, bergantung pada berapa banyak teks yang diinginkan oleh pengguna. Selain teks, Kelas `XMLPage` juga menampung kelas `XMLHolder`. Penjelasan lebih detail mengenai kelas `XMLHolder` akan dijelaskan pada

subbab berikutnya. Dalam satu Kelas XMLPage terdapat 0...n objek XMLHolder. Kelas ini juga terdiri dari *attribute* page untuk menentukan nomor halaman dan UUID sebagai penanda unik pada masing-masing halaman. Susunan atribut dan fungsi yang ada pada Kelas Page dapat dilihat pada Gambar 3.7.

3.2.2.4 Kelas XMLHolder



Gambar 3.8 Kelas XMLHolder

Kelas XMLHolder merupakan kelas yang menampung objek frame dan foto yang didapat dari kelas XMLFrameholder dan XMLPhotoholder. Kelas XMLHolder hanya memiliki 0..1 objek dari dua kelas tersebut. Kelas ini menampung objek yang saling berpengaruh dari kelas XMLPhotoholder dan XMLFrameholder. Susunan atribut dan fungsi yang ada pada Kelas XMLHolder dapat dilihat pada Gambar 3.8.

3.2.2.5 Kelas XMLFrameholder

Kelas XMLFrameholder menampung informasi frame pada aplikasi. Informasi objek yang ditampung dalam bentuk atribut antara lain nama, koordinat x, koordinat y, panjang, lebar, serta gambar frame yang akan digunakan. Informasi gambar yang dalam bentuk atribut gambar. Pada setiap objek Kelas XMLFrameholder terdapat atribut uuid sebagai penanda unik masing-masing objek

dari Kelas XMLFrameholder. Susunan atribut dan fungsi yang ada pada Kelas XMLFrameholder dapat dilihat pada Gambar 3.9.

XMLFrameholder
-name
-uuid
-w
-h
-x
-y
-gambar
+getName()
+setName()
+getUUID()
+setUUID()
+getW()
+setW()
+getH()
+setH()
+getX()
+setX()
+getY()
+setY()
+getImage()
+setImage()

Gambar 3.9 Kelas XMLFrameholder

3.2.2.6 Kelas XMLPhotoholder

Kelas XMLPhotoholder dapat dilihat pada Gambar 3.10 menampung informasi foto pada aplikasi. Informasi yang ditampung antara lain nama, koordinat x, koordinat y, panjang, lebar, serta gambar foto yang akan digunakan. Informasi gambar yang ditampung dalam bentuk elemen.. Susunan atribut dan fungsi yang ada pada Kelas XMLPhotoholder dapat dilihat pada Gambar 3.10.

XMLPhotoholder
-name
-uuid
-w
-h
-x
-y
-gambar
+getName()
+setName()
+getUUID()
+setUUID()
+getW()
+setW()
+getH()
+setH()
+getX()
+setX()
+getY()
+setY()
+getImage()
+setImage()

Gambar 3.10 Kelas XMLPhotoholder

3.2.2.7 Kelas XMLParagraph

Kelas XMLParagraph menampung informasi teks yang telah dibuat maupun akan dibaca oleh aplikasi. Informasi yang ditampung dalam atribut antara lain koordinat x , koordinat y , warna huruf yang direpresentasikan dalam komposisi warna merah, hijau, dan biru, *style* huruf, jenis huruf, serta ukuran huruf yang digunakan dalam proyek tersebut. Kardinalitas objek dari Kelas XMLParagraph pada *file* proyek adalah $0..n$, di mana setiap halaman dapat menampung 0 sampai n buah teks. Susunan atribut dan fungsi yang ada pada Kelas XMLParagraph dapat dilihat pada Gambar 3.11.

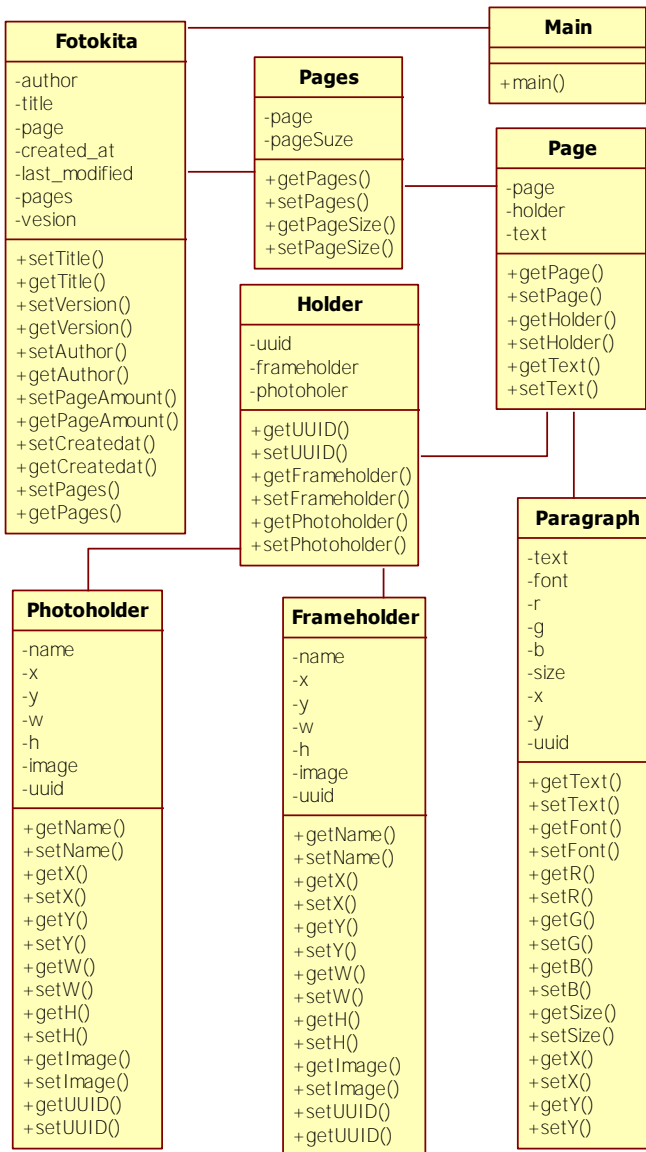
XMLParagraph
-x
-y
-size
-r
-g
-b
-text
-uuid
+getX()
+setX()
+getY()
+setY()
+getSize()
+setSize()
+getR()
+setR()
+getG()
+setG()
+getB()
+setB()
+getText()
+setText()
+getUUID()
+setUUID()

Gambar 3.11 Kelas Paragraph

3.2.3 Perancangan Skema Validasi

Skema validasi digunakan untuk memvalidasi *file* project yang akan dibaca atau akan dibuat agar sesuai dengan standar penggunaan *metadata*. Validator memvalidasi struktur *metadata* dari aplikasi. Kegunaan validasi berdasarkan skema ini adalah agar tidak ada *file* yang tidak sesuai dengan struktur informasi yang tertulis pada *file* proyek .fbk.

Skema validasi digunakan pada setiap kali proyek baru disimpan, serta sesaat sebelum file dibuka. Skema validasi akan berubah sesuai dengan ketentuan dari aplikasi Fotokita, karena proses validasi yang digunakan pada setiap versi aplikasi harus sesuai dengan aplikasi.



Gambar 3.12 Kelas Diagram Skema Validasi XML

Susunan kelas skema validasi pada Gambar 3.12 dijelaskan pada poin-poin berikut.

3.2.3.1 Kelas Fotokita

Fotokita
-author
-title
-page
-created_at
-last_modified
-pages
-vesion
+setTitle()
+getTitle()
+setVersion()
+getVersion()
+setAuthor()
+getAuthor()
+setPageAmount()
+getPageAmount()
+setCreatedat()
+getCreatedat()
+setPages()
+getPages()

Gambar 3.13 Kelas Fotokita

Kelas Fotokita memuat validasi terhadap elemen data author, title, page, created_at, last_modified, pages, dan atribut version. Elemen tersebut merupakan elemen utama penyusun model data pada aplikasi Fotokita. Susunan atribut dan fungsi yang ada pada Kelas Fotokita dapat dilihat pada Gambar 3.13.

3.2.3.2 Kelas Pages

Pages
-page -pageSize +getPages() +setPages() +getPageSize() +setPageSize()

Gambar 3.14 Kelas Pages

Kelas Pages memuat validasi terhadap tipe data elemen page dan size. Susunan atribut dan fungsi yang ada pada Kelas Pages dapat dilihat pada Gambar 3.14.

3.2.3.3 Kelas Page

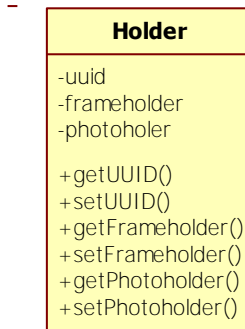
Page
-page -holder -text +getPage() +setPage() +getHolder() +setHolder() +getText() +setText()

Gambar 3.15 Kelas Page

Kelas Page memuat validasi terhadap tipe data elemen pagenumber, holder, dan text. Susunan atribut dan fungsi yang ada pada Kelas Page dapat dilihat pada Gambar 3.15.

3.2.3.4 Kelas Holder

Kelas Holder memuat validasi terhadap tipe data elemen holder, yang menjadi *parent* dari elemen frameholder dan photoholder. Susunan atribut dan fungsi yang ada pada Kelas Holder dapat dilihat pada Gambar 3.16.



Gambar 3.16 Kelas Holder

3.2.3.5 Kelas Frameholder

Kelas Frameholder memuat validasi terhadap tipe data atribut name, x, y, w, h, dan *value* image pada elemen frameholder. Susunan atribut dan fungsi yang ada pada Kelas Frameholder dapat dilihat pada Gambar 3.17.

3.2.3.6 Kelas Photoholder

Kelas Photoholder memuat validasi terhadap tipe data atribut name, x, y, w, h, dan *value* image pada *element* photoholder. Susunan atribut dan fungsi yang ada pada Kelas Photoholder dapat dilihat pada Gambar 3.18.

-

Frameholder
-name -x -y -w -h -Image +getName() +setName() +getX() +setX() +getY() +setY() +getW() +setW() +getH() +setH() +getImage() +setImage()

Gambar 3.17 Kelas Frameholder

-

Photoholder
-name -x -y -w -h -image +getName() +setName() +getX() +setX() +getY() +setY() +getW() +setW() +getH() +setH() +getImage() +setImage()

Gambar 3.18 Kelas Photoholder

3.2.3.7 Kelas Paragraph

Paragraph
-text
-font
-color
-style
-size
+getText()
+setText()
+getFont()
+setFont()
+getColor()
+setColor()
+getStyle()
+setStyle()
+getSize()
+setSize()

Gambar 3.19 Kelas Paragraph

Kelas Paragraph memuat validasi terhadap tipe data *element text*, *font*, *color*, *style*, dan *size*. Susunan atribut dan fungsi yang ada pada Kelas Paragraph dapat dilihat pada Gambar 3.19.

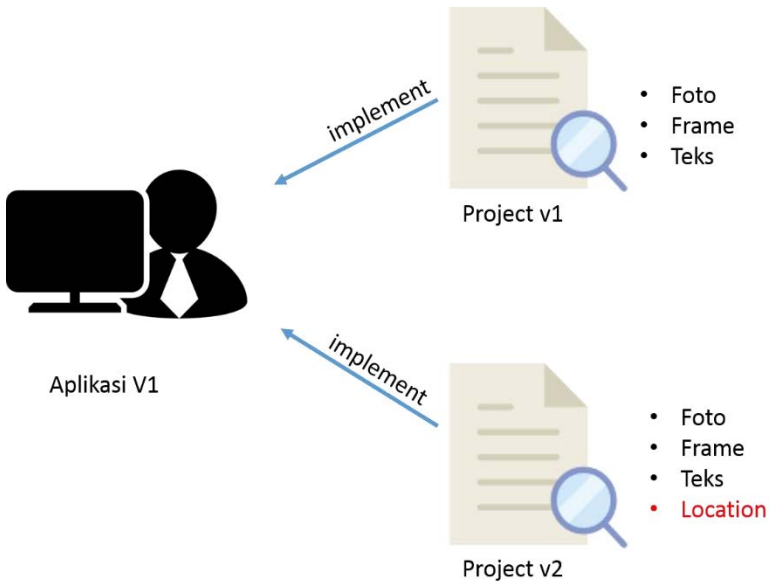
3.2.4 Perancangan Skema Forward dan Backward Compatibility

Perancangan skema kompatibilitas aplikasi pada setiap versi diatur dalam skema validasi. Dalam skema tersebut terdapat dua kasus, yaitu *Forward Compatibility* dan *Backward Compatibility*.

3.2.4.1 Forward Compatibility

Perancangan skema *forward compatibility* mengakomodasi kemungkinan sebuah aplikasi versi lebih rendah membaca *file* proyek yang disimpan dalam versi lebih tinggi. Karena sistem aplikasi versi lebih rendah tidak memiliki fitur seperti sistem

aplikasi versi lebih tinggi, maka ada skema *forward compatibility* mengantisipasi kesalahan keadaan tersebut dengan tidak mengimplementasi fitur dari *file* proyek versi lebih tinggi yang tidak ada pada sistem aplikasi versi lebih rendah. Keadaan tersebut dapat dilihat pada Gambar 3.20.

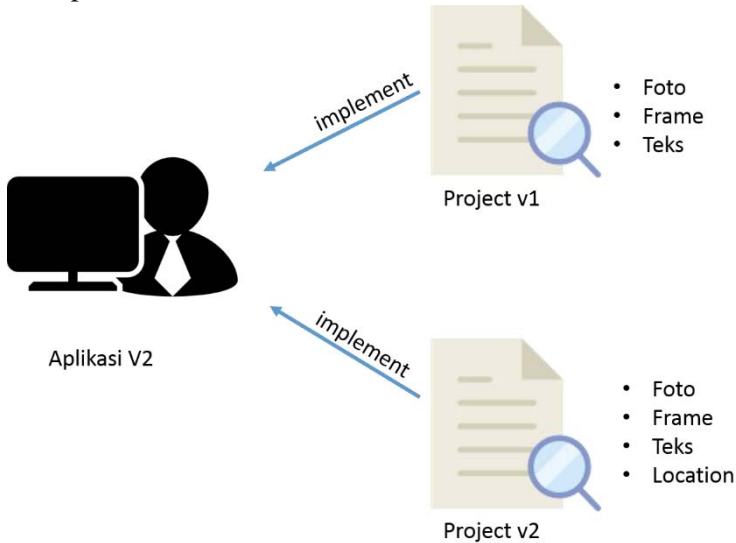


Gambar 3.20 Perancangan Skema Forward Compatibility

3.2.4.2 Backward Compatibility

Perancangan skema *backward compatibility* mengakomodasi kemungkinan sebuah aplikasi versi lebih tinggi membaca *file* proyek yang disimpan dalam versi lebih rendah. Pada kasus ini, sistem aplikasi baru memiliki fitur yang lebih banyak dari *file* proyek versi lebih rendah. Karena sistem aplikasi versi lebih tinggi dapat mengakomodasi fitur yang terdapat pada *file* proyek versi lebih rendah, maka sistem aplikasi versi lebih tinggi dapat mengimplementasi seluruh fitur yang terdapat pada *file*

proyek versinya maupun lebih rendah. Keadaan tersebut dapat dilihat pada Gambar 3.21.



Gambar 3.21 Perancangan Skema Backward Compatibility

3.2.5 Perancangan *Partial Update*

Partial Update yang dirancang untuk tugas akhir ini adalah dengan meminimalisir penggunaan *memory* pada saat penyimpanan. Untuk *file* yang berubah, maka tag yang diubah hanya elemen yang berganti nilai. Elemen-elemen yang tidak diubah tidak ditimpa, sehingga saat proses penyimpanan nilai elemen tersebut adalah nilai elemen yang dibaca saat aplikasi membuka *file*.

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem sesuai dengan analisis dan perancangan sistem penyimpanan pada aplikasi Fotokita secara umum.

Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, kode sumber utama dan implementasi *library* dan *template* yang digunakan dalam pengerjaan tugas akhir. Bahasa pemrograman yang digunakan adalah Java dengan JAXB guna memudahkan pembuatan *metadata*.

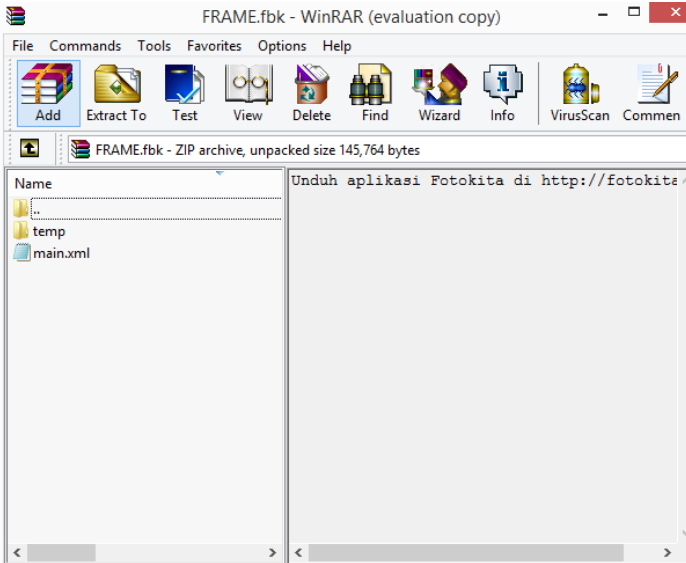
4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi pada tugas akhir ini dilakukan pada lingkungan dengan kaskas sebagai berikut.

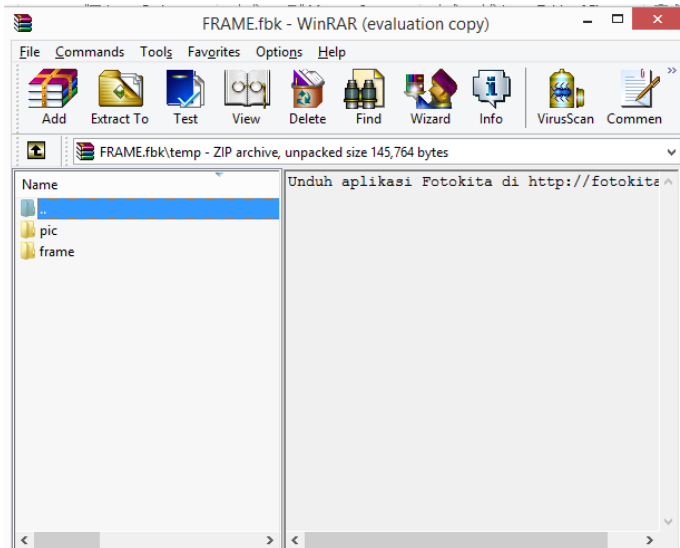
1. Sistem operasi Windows Embedded 8.1 Industry Pro 64 bit.
2. Eclipse Luna digunakan sebagai editor kode.
3. StarUML 2 digunakan untuk membuat diagram kelas.
4. Google Chrome digunakan untuk melakukan pengecekan terhadap *file* berekstensi *.fbk* yang dihasilkan dari aplikasi Fotokita.

4.2 Implementasi Struktur File

Struktur file yang diimplementasikan sesuai dengan perancangan ada pada Gambar 4.1 dan Gambar 4.2 Gambar 4.1 menunjukkan struktur utama file *.fbk* yang berisi file *main.xml* serta folder *temp*. Folder *temp* berisi foto dan frame berupa *file binary* yang akan diimplementasikan secara langsung pada aplikasi editor Fotokita.



Gambar 4.1 Struktur objek penyusun file .fbk



Gambar 4.2 Struktur objek elemen pada file .fbk

4.3 Penggunaan Masing-masing Kelas

Pada bab ini dibahas mengenai implementasi dari perancangan diagram kelas yang telah dijelaskan pada bab III.

4.3.1 Penggunaan Kelas Pada XML

Penggunaan kelas pada XML agar dapat menyimpan dan membuka *file metadata*. Kelas-kelas yang diimplementasikan dalam proses menyimpan dan membaca *file metadata* adalah sebagai berikut.

4.3.1.1 Penggunaan Kelas Fotokita

Kelas Fotokita merupakan kelas utama yang mengatur *root element* pada *metadata* dalam bentuk *tag* `<fotokita>`. Selain itu kelas ini mengakomodasi *set* dan *get* atribut versi `'version'` untuk menyimpan elemen versi. Implementasi lain diantaranya atribut judul yang diimplementasikan dengan elemen `<title>`, pemilik proyek `<author>`, jumlah halaman `<pageamount>`, waktu pembuatan `<createdat>`, waktu perubahan `<lastmodified>`, serta termasuk fungsi *set* dan *get* Kelas Pages.

4.3.1.2 Penggunaan Kelas Pages

Kelas Pages memuat informasi ukuran dalam bentuk elemen `<size>`, serta menampung informasi halaman-halaman yang dibuat dalam aplikasi ke dalam *list* `<page>`.

4.3.1.3 Penggunaan Kelas Page

Pada Kelas Page dimuat informasi mengenai setiap halaman. Informasi yang dimuat antara lain nomor halaman, teks, frame, serta gambar foto yang ada dalam halaman. Dalam kelas ini terdapat fungsi *setter* dan *getter* untuk seluruh *tag* dan atribut. Informasi nomor halaman tersimpan dalam atribut `'no'`, sementara untuk fragmen yang menyimpan gambar foto disimpan menjadi satu kesatuan dalam elemen `<holder>`, dan elemen teks disimpan dalam bentuk `<text>`.

4.3.1.4 Penggunaan Kelas Holder

Implementasi Kelas Holder adalah pada elemen *tag* `<holder>`, di mana elemen tersebut merupakan elemen pembungkus elemen yang dihasilkan oleh Kelas Frameholder dan Kelas Photoholder. Elemen ini mempunyai atribut `uuid` sebagai penanda unik.

4.3.1.5 Penggunaan Kelas Frameholder

Kelas Frameholder memuat informasi frame yang digunakan pada setiap halamannya. Di dalam kelas ini terdapat fungsi untuk *setter* dan *getter* atribut nama, `uuid`, posisi `x`, `y`, panjang dan lebar frame, serta terdapat informasi gambar.

4.3.1.6 Penggunaan Kelas Photoholder

Kelas Photoholder memuat informasi foto yang digunakan pada setiap halamannya. Di dalam kelas ini terdapat fungsi untuk *setter* dan *getter* atribut nama, `uuid`, posisi `x`, `y`, panjang dan lebar foto, serta terdapat informasi gambar.

4.3.1.7 Penggunaan Kelas Paragraph

Kelas Paragraph memuat informasi teks yang digunakan dalam setiap halaman. Terdapat fungsi setter dan getter atribut font, warna teks, ukuran teks, bentuk teks, serta posisi `x` dan `y` dari teks.

4.3.1.8 Penggunaan Kelas MyValidationEventHandler

Kelas `MyValidationEventHandler` untuk menindaklanjuti kesalahan yang muncul ketika proyek *file* yang dibuka tidak memenuhi skema *file* yang ditentukan oleh sistem aplikasi. Potongan kode sumber dari implementasi kelas validasi dapat dilihat pada Kode Sumber 4.1.

```
public boolean handleEvent(ValidationEvent
event) {
    System.out.println("MESSAGE: " +
event.getMessage());
    int style = SWT.ICON_ERROR;
```

```
        MessageBox messageBox = new
MessageBox(MainWindow.getInstance(), style);
        messageBox.setMessage("Aplikasi
membutuhkan versi yang lebih baru. Silakan
download di http://fotokita.com");
        int rc = messageBox.open();

        switch (rc) {
        case SWT.OK:
            System.out.println("SWT.OK");
            break;
        case SWT.CANCEL:
System.out.println("SWT.CANCEL");
            break;
        case SWT.YES:
            System.out.println("SWT.YES");
            break;
        case SWT.NO:
            System.out.println("SWT.NO");
            break;
        case SWT.RETRY:
            System.out.println("SWT.RETRY");
            break;
        case SWT.ABORT:
            System.out.println("SWT.ABORT");
            break;
        case SWT.IGNORE:
System.out.println("SWT.IGNORE");
            break;
        }

        return true;
    }
}
```

Kode Sumber 4.1 Implementasi Fungsi Kelas Validasi

4.3.2 Penggunaan Kelas pada Skema Validasi XML

Penggunaan kelas pada skema validasi XML diimplementasikan untuk membuat susunan skema yang digunakan sebagai validator *file* objek .fbk. Penjelasan mengenai kelas yang ada pada skema validasi XML tersebut dijelaskan lebih lanjut pada subbab berikut.

4.3.2.1 Penggunaan Kelas Fotokita

Kelas Fotokita merupakan kelas utama yang mengatur root *element* pada *metadata* dalam bentuk tag “<fotokita>”. Selain itu kelas ini mengakomodasi *set* dan *get* atribut versi “<version>” dan beberapa elemen, diantaranya elemen judul “<title>”, pemilik proyek “<author>”, jumlah halaman “<page>”, waktu pembuatan “<createdat>”, waktu pengubahan “<lastmodified>”, serta termasuk fungsi *set* dan *get* pada Kelas Pages.

4.3.2.2 Penggunaan Kelas Pages

Kelas Pages memuat informasi ukuran dalam bentuk atribut *size* “<size>”, serta menampung informasi halaman-halaman yang dibuat dalam aplikasi ke dalam *list* “<page>”.

4.3.2.3 Penggunaan Kelas Page

Pada Kelas Page dimuat informasi mengenai setiap halaman. Informasi yang dimuat antara lain nomor halaman, teks, frame, serta gambar foto yang ada dalam halaman. Dalam kelas ini terdapat fungsi *setter* dan *getter* untuk seluruh *tag* dan atribut. Informasi nomor halaman tersimpan dalam attribute “no”, sementara untuk fragmen yang menyimpan gambar foto disimpan dalam bentuk elemen “<photoholder>”. Elemen frame disimpan dalam bentuk “<frameholder>”, dan elemen teks disimpan dalam bentuk “<text>”.

4.3.2.4 Penggunaan Kelas Frameholder

Kelas Frameholder memuat informasi frame yang digunakan pada setiap halamannya. Di dalam kelas ini terdapat

fungsi untuk setter dan getter atribut nama, uuid, posisi x, y, panjang dan lebar frame, serta terdapat informasi gambar.

4.3.2.5 Penggunaan Kelas Photoholder

Kelas Photoholder memuat informasi foto yang digunakan pada setiap halamannya. Di dalam kelas ini terdapat fungsi untuk setter dan getter atribut nama, uuid, posisi x, y, panjang dan lebar foto, serta terdapat informasi gambar.

4.3.2.6 Penggunaan Kelas Paragraph

Kelas Paragraph memuat informasi teks yang digunakan dalam setiap halaman. Terdapat fungsi setter dan getter atribut font, warna teks, ukuran teks, bentuk teks, serta posisi x dan y dari teks.

4.4 Implementasi Fungsi Tulis dan Baca XML menggunakan JAXB dan DOM

Fungsi utama pada JAXB adalah untuk menulis (*marshal*) dan membaca (*unmarshal*) file proyek ke dalam bentuk XML. Dua fungsi utama JAXB tersebut dijelaskan lebih lanjut.

4.4.1 Fungsi Marshalling JAXB

Fungsi `marshalling()` mengimplementasi JAXB dan SAX. Objek yang akan disimpan dalam bentuk dokumen XML terlebih dahulu dicek dengan skema XML yang ada. Pengecekan ini bertujuan untuk memastikan tidak adanya proyek XML yang tidak sesuai dengan skema XML yang telah ditentukan. Kode sumber fungsi marshalling disertakan dalam Kode Sumber 4.2.

```
public static void marshalling() throws
JAXBException, SAXException
{
    File file = new File(path
    JAXBContext jaxbContext =
    JAXBContext.newInstance(Fotokita.class);

    Marshaller jaxbMarshaller =
    jaxbContext.createMarshaller();
```

```

SchemaFactory sf =
SchemaFactory.newInstance(XMLConstants.W
3C_XML_SCHEMA_NS_URI);
Schema schema = sf.newSchema(new
File("/Project/COBATA/fotokitaxsd.xml"))
;

jaxbMarshaller.setSchema(schema);
jaxbMarshaller.setProperty(Marshaller.JA
XB_FORMATTED_OUTPUT, true);
jaxbMarshaller.marshal(xml, file);
}

```

Kode Sumber 4.2 Implementasi Fungsi Tulis (Marshalling)

4.4.2 Fungsi Unmarshalling menggunakan DOM

Fungsi unmarshalling() mengimplementasi DOM dan SAX. Objek yang akan dibuka dari bentuk dokumen XML terlebih dahulu dicek dengan skema XML yang ada. Pengecekan ini bertujuan untuk memastikan tidak adanya proyek XML yang tidak sesuai dengan skema XML yang telah ditentukan. Kode sumber fungsi unmarshalling() disertakan dalam Kode Sumber 4.3.

```

public static void unmarshalling(final Display
display) throws SAXException, IOException,
XPathExpressionException,
ParserConfigurationException, JAXBException {
    DocumentBuilderFactory dbf1 =
DocumentBuilderFactory.newInstance();
    DocumentBuilder docBuilder1 =
dbf1.newDocumentBuilder();
    Document fotokita_xml =
docBuilder1.parse("/Project/COBATA/fotokita100
.fbk");

    SchemaFactory schemaFactory =
SchemaFactory.newInstance(XMLConstants.W3C_XML
SCHEMA_NS_URI);
}

```

```

        cekSchema();
    }

```

Kode Sumber 4.3 Penggunaan Fungsi Unmarshalling

4.4.3 Penggunaan Fungsi cekSchema()

Fungsi `cekSchema()` digunakan untuk mengecek apakah *file* XML yang ada sudah sesuai dengan skema XML. Jika *file* dokumen XML berbeda, maka akan dimunculkan pesan peringatan. Kode sumber `cekSchema` disertakan dalam Kode Sumber 4.4.

```

public static void cekSchema() throws
SAXException, JAXBException{
    SchemaFactory schemaFactory =
SchemaFactory.newInstance(XMLConstants.W3C_XML
_SCHEMA_NS_URI);
    Schema schema =
schemaFactory.newSchema(new
File("/Project/COBATA/fotokitaxsd.xsd"));

    JAXBContext xpath_jc =
JAXBContext.newInstance(Fotokita.class);

    Unmarshaller unmarshaller =
xpath_jc.createUnmarshaller();
        unmarshaller.setSchema(schema);
        unmarshaller.setEventHandler(new
MyValidationEventHandler());
        try {
            Fotokita fotokita xml schema=
(Fotokita) unmarshaller.unmarshal(new
File("/Project/COBATA/fotokita100.fbK"));
        } catch (JAXBException e1) {
            System.out.println("BEDA");
        }
    }
}

```

Kode Sumber 4.4 Penggunaan Fungsi cekSchema

4.5 Penggunaan Library dan Template

Dalam pengerjaan tugas akhir ini digunakan beberapa *Library* dan *template* yang membantu memudahkan pembuatan sistem penyimpanan aplikasi ini.

4.5.1 Penggunaan Library UUID

Library UUID digunakan sebagai penanda masing-masing elemen *metadata*, untuk memudahkan melakukan perubahan pada elemen tertentu. Penggunaan *Library* UUID dapat dilihat pada Kode Sumber 4.5.

```
import java.util.UUID;

@XmlAttribute(name = "uuid")
public UUID getUUID() {
    return uuid;
}
public void setUUID(UUID uuid) {
    this.uuid = uuid;
}
```

Kode Sumber 4.5 Penggunaan Library UUID

4.5.2 Penggunaan Library JAXB

Implementasi *library* JAXB digunakan pada setiap *member* pada kelas untuk menentukan bagian *member* kelas yang berfungsi sebagai elemen dan anggota kelas yang berfungsi sebagai atribut ataupun *value* pada dokumen XML. Beberapa contoh implementasi *attribute*, *element*, dan *value* pada suatu *node* dituliskan pada Kode Sumber 4.6, Kode Sumber 4.7, dan Kode Sumber 4.8.

```
@XmlAttribute(name="version")
public String getVersion()
{
    return versi;
}
public void setVersion(String versi) {
```

```

    this.versi = versi;
}

```

Kode Sumber 4.6 Implementasi JAXB pada Atribut “version”

```

@XmlElement(name="title")
public String getTitle()
{
    return judul;
}
public void setTitle(String judul){
    this.judul = judul;
}

```

Kode Sumber 4.7 Implementasi JAXB pada Elemen “title”

```

@XmlValue()
public String getText() {
    return text;
}
public void setText(String text) {
    this.text = text;
}

```

Kode Sumber 4.8 Implementasi JAXB pada Value

Implementasi penggunaan library JAXB secara detail dijelaskan pada Tabel 4.1.

Tabel 4.1 Implementasi JAXB pada Dokumen XML

Kelas	Member	Implementasi	Tag name
Fotokita	versi	<i>Attribute</i>	version
	auth	<i>Element</i>	author
	created	<i>Element</i>	createdat
	modified	<i>Element</i>	lastmodified
	page	<i>Element</i>	pageamount
	judul	<i>Element</i>	title
Pages	pageSize	<i>Attribute</i>	size

Kelas	Member	Implementasi	Tag name
	halaman	<i>Element</i>	page
Page	page	<i>Attribute</i>	no
	text	<i>Element</i>	text
	holder	<i>Element</i>	holder
Holder	Photoholder	<i>Element</i>	Photoholder
	FrameHolder	<i>Element</i>	Frameholder
Photoholder	name	<i>Attribute</i>	name
	x	<i>Attribute</i>	x
	y	<i>Attribute</i>	y
	w	<i>Attribute</i>	w
	h	<i>Attribute</i>	h
	gambar	<i>Element</i>	image
Frameholder	name	<i>Attribute</i>	name
	x	<i>Attribute</i>	x
	y	<i>Attribute</i>	y
	w	<i>Attribute</i>	w
	h	<i>Attribute</i>	h
	gambar	<i>Element</i>	image
Paragraph	text	<i>value</i>	text
	font	<i>Attribute</i>	font
	color	<i>Attribute</i>	color
	size	<i>Attribute</i>	size
	x	<i>Attribute</i>	x
	y	<i>Attribute</i>	y
	<i>style</i>	<i>Attribute</i>	style

4.6 Implementasi Forward dan Backward Compatibility

Implementasi *forward compatibility* dan *backward compatibility* dijelaskan pada subbab berikut.

4.6.1 Implementasi Forward Compatibility

Implementasi penggunaan *forward compatibility* dilakukan dengan menambah elemen pada *file* berekstensi *.fbk* yang telah dibuat. Penambahan elemen baru adalah elemen *location* yang berada pada elemen anakan dari elemen *fotokita*. Penambahan tersebut memiliki struktur yang sama dengan elemen yang ada pada versi lebih tinggi.

4.6.2 Implementasi Backward Compatibility

Implementasi penggunaan *backward compatibility* dilakukan dengan menyimpan *file* proyek yang tidak mengimplementasikan elemen teks. Pengembangan versi lebih rendah tidak mempunyai elemen pada struktur model *metadata*-nya.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan pengujian dan evaluasi dari penyimpanan proyek pada aplikasi Fotokita. Pengujian yang dilakukan adalah pengujian dalam bentuk *control test* yang diselenggarakan pada waktu yang ditentukan. Pengujian memperhatikan terpenuhinya kebutuhan fungsionalitas dan konsistensi dari sistem yang sesuai dengan skenario.

5.1 Lingkungan Pengujian

Lingkungan pengujian adalah lingkungan, baik perangkat keras maupun perangkat lunak tempat pengujian sistem dilakukan. Pengujian untuk rancang bangun penyimpanan dinamis dilakukan pada komputer dengan spesifikasi pada Tabel 5.1.

Tabel 5.1 Spesifikasi Pengujian

Spesifikasi	Deskripsi
CPU	Intel ® Core™ i7-3612QM CPU @ 2.10 GHz
RAM	8.00 GB
Sistem Operasi	Windows 8.1 Embedded Industry Pro 64 bit

Pengujian dilakukan dengan membuat *file* proyek. Pengujian berhasil apabila *file* yang disimpan atau dibaca sama dengan hasil yang ditampilkan pada aplikasi Fotokita.

5.2 Skenario Pengujian Fungsionalitas

Pada subbab ini dijelaskan mengenai skenario pengujian fungsionalitas yang digunakan. Seperti yang sudah dijelaskan pada pengantar di bab 5, bahwa pengujian merupakan pengujian fungsionalitas yang menggunakan metode *blackbox*. Metode *blackbox* merupakan metode dimana pengujian ditekankan pada pola *input* dan *output* yang sesuai dengan skenario. Pengujian

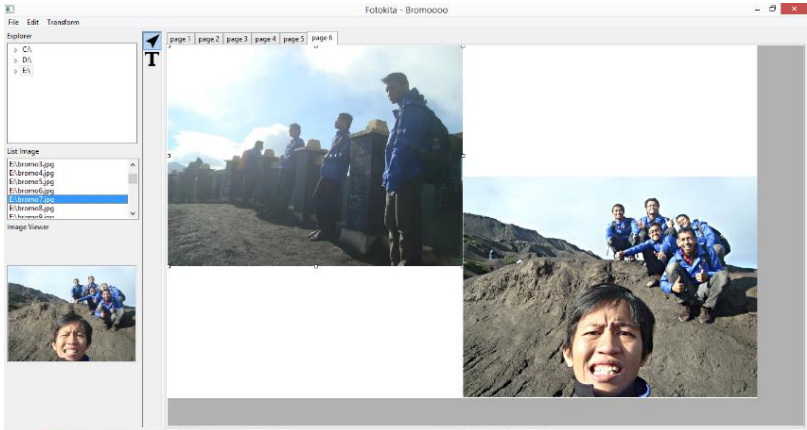
dilakukan dengan mengacu kasus penggunaan yang dijelaskan pada bab 3, analisis dan perancangan sistem. Untuk menguji konsistensi aplikasi, pengujian dilakukan sebanyak 10 kali.

5.2.1 Pengujian Menyimpan File Proyek

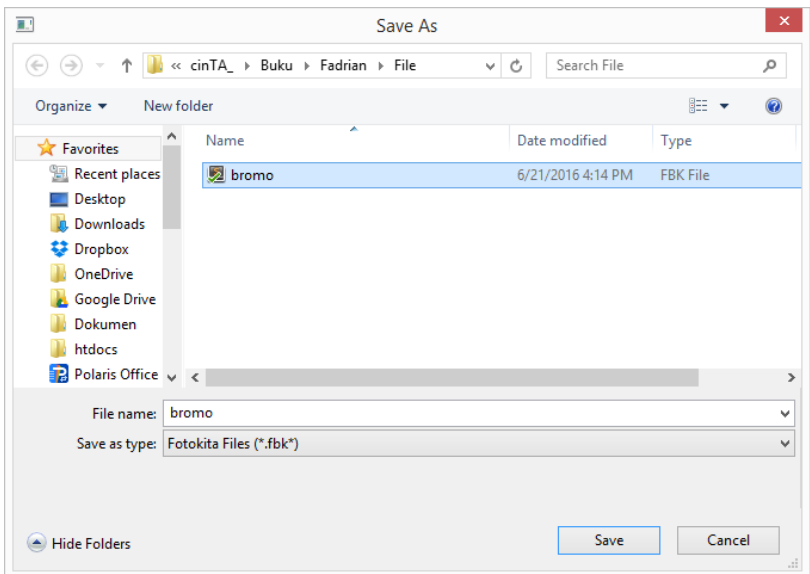
Pada pengujian ini pengguna yang telah membuat proyek baru menyimpan proyek tersebut ke dalam *file* berekstensi .fbk ke dalam media penyimpanan internal komputer. Skenario pengujian menyimpan *file* proyek terdapat pada Tabel 5.2. Hasil pengujian dapat dilihat pada Gambar 5.1, Gambar 5.2, dan Gambar 5.3.

Tabel 5.2 Skenario Uji Coba Menyimpan File Proyek

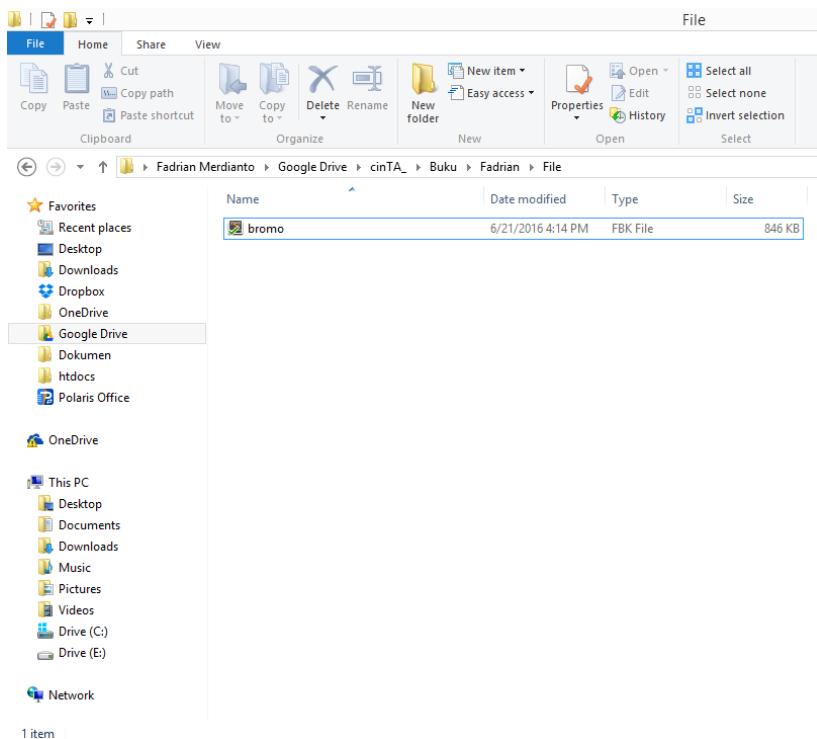
ID	UJ-001
Kasus Penggunaan	Menyimpan file proyek
Nama	Menyimpan file proyek
Tujuan Pengujian	Menguji fitur penyimpanan file proyek pada aplikasi
Skenario	Pengguna menyimpan file proyek yang telah dibuat
Kondisi Awal	Pengguna telah membuat proyek dan belum tersimpan
Langka Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuat proyek baru 2. Pengguna menyimpan proyek dalam bentuk <i>file</i> .fbk
Kondisi yang diharapkan	Proyek tersimpan dalam <i>file</i> dengan format .fbk
Kondisi yang didapat	<i>File</i> hasil proyek tersimpan dalam format .fbk
Hasil Pengujian	Berhasil



Gambar 5.1 Hasil Proyek yang Telah Dibuat



Gambar 5.2 Dialog Penyimpanan File Proyek



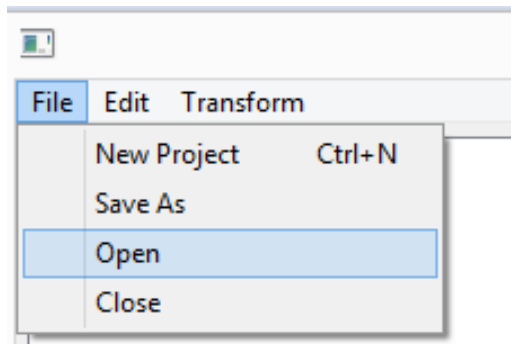
Gambar 5.3 File Hasil Penyimpanan

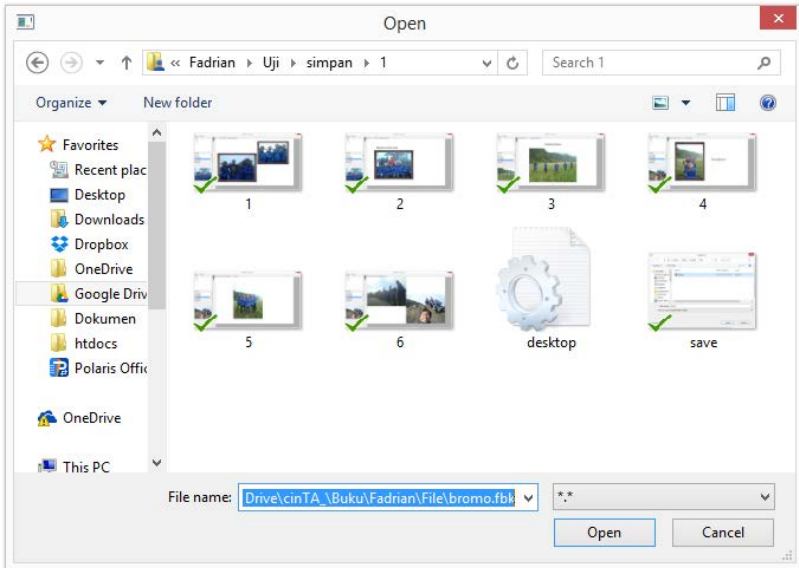
5.2.2 Pengujian Membaca File Proyek dengan Versi Aplikasi yang sama

Pada pengujian ini pengguna yang telah menyimpan *file* proyek dalam bentuk *.fbk* membuka kembali *file* tersebut dari penyimpanan internal komputer lalu digambarkan kembali ke dalam aplikasi editor. Skenario pengujian membaca *file* proyek dengan versi aplikasi yang sama terdapat pada Tabel 5.3. Hasil pengujian dapat dilihat pada Gambar 5.4 dan Gambar 5.5.

Tabel 5.3 Skenario Uji Coba Membaca File Proyek

ID	UJ-002
Kasus Penggunaan	Membaca file proyek
Nama	Membaca file proyek
Tujuan Pengujian	Menguji fitur penbacaan file proyek pada aplikasi
Skenario	Pengguna membuka proyek yang telah dibuat dengan aplikasi
Kondisi Awal	Pengguna telah mempunyai file proyek dalam bentuk ekstensi .fbk
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>Open</i> 2. Pengguna memilih <i>file</i> berekstensi .fbk 3. Pengguna membuka file .fbk
Kondisi diharapkan	yang Hasil gambar pada <i>file</i> proyek muncul pada tampilan aplikasi
Kondisi didapat	yang Aplikasi mampu membaca file proyek dengan mengimplementasi elemen yang ada
Hasil Pengujian	Berhasil

**Gambar 5.4 Dialog Membuka File Proyek**



Gambar 5.5 Tampilan Hasil File Proyek yang Telah dibaca

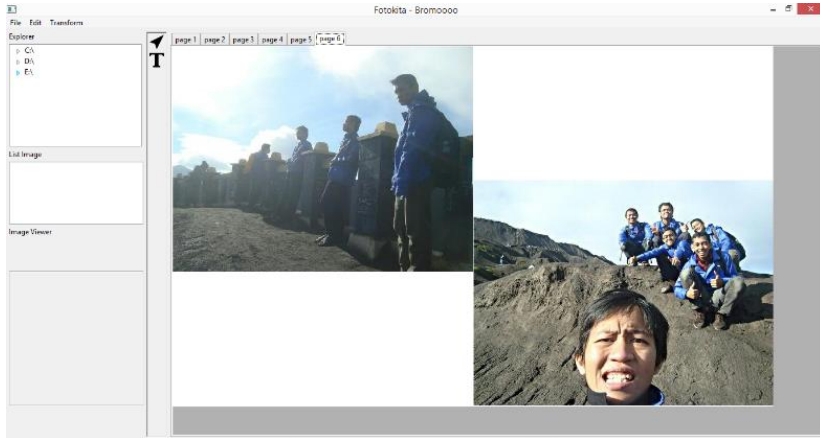
5.2.3 Pengujian Menyimpan File Proyek yang Telah diubah

Pada pengujian ini pengguna yang telah membuka *file* proyek dalam bentuk .fbk menyimpan kembali *file* tersebut setelah diubah. Skenario menyimpan file proyek yang telah diubah terdapat pada Tabel 5.4. Hasil pengujian dapat dilihat pada Gambar 5.6, Gambar 5.7, Gambar 5.8 dan Gambar 5.9.

Tabel 5.4 Skenario Uji Coba Menyimpan File Proyek yang telah diubah

ID	UJ-003
Kasus Penggunaan	Menyimpan file proyek yang telah diubah
Nama	Menyimpan file proyek yang telah diubah

Tujuan Pengujian	Menguji fitur penyimpanan file proyek yang berasal dari proyek yang dibaca
Skenario	Pengguna menyimpan kembali file proyek yang telah diubah dari file proyek yang dibaca
Kondisi Awal	Pengguna telah mempunyai <i>file</i> proyek dalam bentuk ekstensi <i>.fbk</i> dan dibuka oleh aplikasi
Langkah Pengujian	<ol style="list-style-type: none"> 1. Penguji membuka <i>file</i> berekstensi <i>.fbk</i> ke dalam aplikasi 2. Penguji melakukan perubahan pada kanvas (menambah/mengurangi/ mengubah elemen) 3. Penguji menyimpan kembali proyek ke dalam <i>file .fbk</i>
Kondisi yang diharapkan	<i>File</i> proyek yang telah diubah tersimpan kembali dalam format <i>.fbk</i>
Kondisi yang didapat	<i>File</i> proyek tersimpan kembali dalam format <i>.fbk</i> dengan susunan informasi <i>metadata</i> sesuai dengan perubahan
Hasil Pengujian	Berhasil



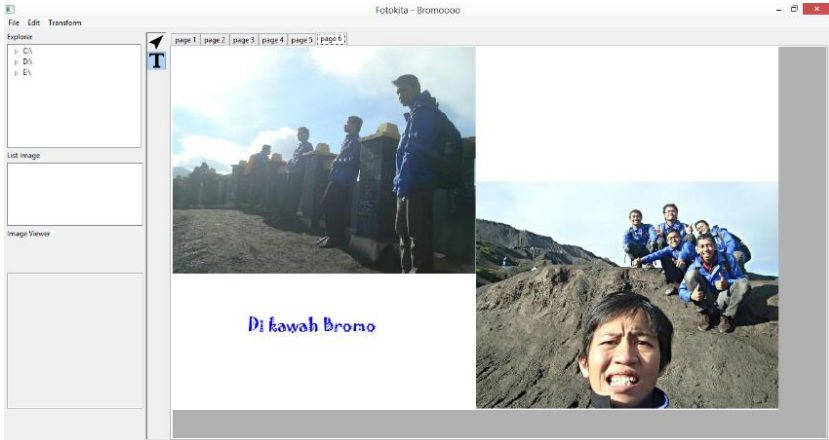
Gambar 5.6 Hasil Proyek yang Telah Dibuka

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <fotokita version="1">
3   <author>Fadri</author>
4   <createdat>2016-06-21T16:08:13.370+07:00</createdat>
5   <lastmodified>2016-06-21T16:14:25.564+07:00</lastmodified>
6   <pageamount>6</pageamount>
7   <pages size="A4">
8     <page no="0" uuid="ab6ed746-87cf-4d27-9cea-62c8c24865b1">
25     </page>
26     <page no="1" uuid="59edcfbd-caf5-468a-9115-ebb11b58f533">
36     </page>
37     <page no="2" uuid="3794fc3d-ba75-4fbe-b261-f573c0abbf17">
45     </page>
46     <page no="3" uuid="93e24007-baae-4883-9a54-4f972996c2fd">
56     </page>
57     <page no="4" uuid="a06907a0-43e1-4bbb-9aa0-17e596b7ec5e">
64     </page>
65     <page no="5" uuid="af5753b4-f894-413a-9739-a4ce0b8ba468">
66       <holder uuid="7f583964-c6be-4f25-82ab-d0cc4ea0bcfb">
71       </holder>
72       <holder uuid="c8b45c20-bbd9-4f6c-9f35-93c70a02de35">
77       </holder>
78     </page>
79   </pages>
80   <title>Bromo000</title>
81   <uuid>936ca21d-5d88-4d13-b35d-12bc6b66361a</uuid>
82 </fotokita>
83

```

Gambar 5.7 Susunan Metadata Ketika dibaca



Gambar 5.8 Proyek Setelah Diubah

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <fotokita version="1">
3   <author>Fadri</author>
4   <createdat>2016-06-21T16:08:13.370+07:00</createdat>
5   <lastmodified>2016-06-21T16:23:04.547+07:00</lastmodified>
6   <pageamount>6</pageamount>
7   <pages size="A4">
8     <page no="0" uuid="ab6ed746-87cf-4d27-9cea-62c8c24865b1">
25     </page>
26     <page no="1" uuid="59edcfbd-caf5-468a-9115-ebb11b58f533">
36     </page>
37     <page no="2" uuid="3794fc3d-ba75-4fbe-b261-f573c0abbf17">
45     </page>
46     <page no="3" uuid="93e24007-baae-4883-9a54-4f972996c2fd">
56     </page>
57     <page no="4" uuid="a06907a0-43e1-4bbb-9aa0-17e596b7ec5e">
64     </page>
65     <page no="5" uuid="af5753b4-f894-413a-9739-a4ce0b8ba468">
66       <holder uuid="7f583964-c6be-4f25-82ab-d0cc4ea0bcfb">
71       </holder>
72       <holder uuid="c8b45c20-bbd9-4f6c-9f35-93c70a02de35">
77       </holder>
78       <text b="255" font="Jokerman" g="0" r="0" size="20" uuid
          = "8fe98a02-50f3-44a6-bf84-7e748e5f2344" . x="124" y="440">
          Di Kawah Bromo</text>
79     </page>
80   </pages>
81   <title>Bromo000</title>
82   <uuid>936ca21d-5d88-4d13-b35d-12bc6b66361a</uuid>
83 </fotokita>

```

Gambar 5.9 Susunan Metadata setelah Proyek diubah

5.2.4 Pengujian Membaca File Proyek dengan Versi Aplikasi yang berbeda

Pengujian ini bertujuan membuktikan aplikasi tetap dapat berjalan walaupun *file* proyek berbeda versi dengan aplikasi yang terpasang. Pengujian ini dibagi menjadi 2, yaitu *forward compatibility* dan *backward compatibility*.

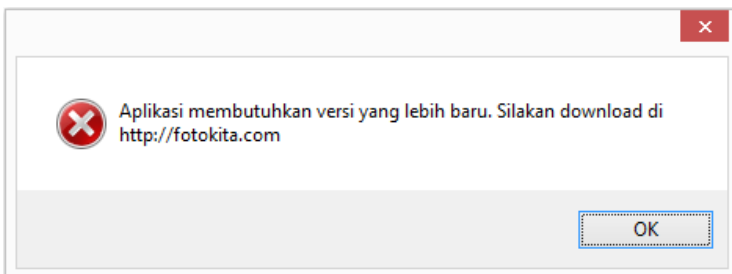
5.2.4.1 Pengujian Forward Compatibility

Pengujian ini bertujuan membuktikan aplikasi dengan versi lama dapat mengakomodasi *file* proyek yang disimpan dalam versi terbaru dengan mengimplementasikan fitur yang ada pada versi lama. Skenario pengujian *forward compatibility* terdapat pada Tabel 5.5. Hasil pengujian dapat dilihat pada Gambar 5.10, Gambar 5.11, dan Gambar 5.12.

Tabel 5.5 Skenario Uji Coba Skema Forward Compatibility

ID	UJ-004
Kasus Penggunaan	Membaca file proyek dengan versi aplikasi terbaru menggunakan aplikasi dengan versi lama
Nama	Forward Compatibility
Tujuan Pengujian	Menguji kemampuan sistem untuk mengakomodasi pembaruan yang dilakukan pada sistem
Skenario	Pengguna membuka file proyek versi lebih tinggi menggunakan aplikasi versi lebih rendah
Kondisi Awal	Menyimpan proyek ke dalam <i>file</i> .fbk
Langkah Pengujian	1. Mengubah struktur <i>metadata file</i> yang telah disimpan dengan aplikasi pengubah teks (menambahkan tag <location> pada proyek)

ID	UJ-004
	2. Menyimpan <i>file</i> .fbk yang telah diubah 3. Membuka <i>file</i> .fbk dengan aplikasi Fotokita
Kondisi yang diharapkan	<ul style="list-style-type: none"> • Aplikasi menampilkan pesan peringatan untuk mengunduh aplikasi dengan versi lebih tinggi • Aplikasi menampilkan proyek ke dalam kanvas dengan mengimplementasikan fitur-fitur yang ada pada aplikasi versi lebih rendah
Kondisi yang didapatkan	<ul style="list-style-type: none"> • Muncul pesan peringatan untuk memperbarui aplikasi • Fitur yang diimplementasikan pada aplikasi adalah fitur yang ada pada aplikasi versi lebih rendah
Hasil Pengujian	Berhasil



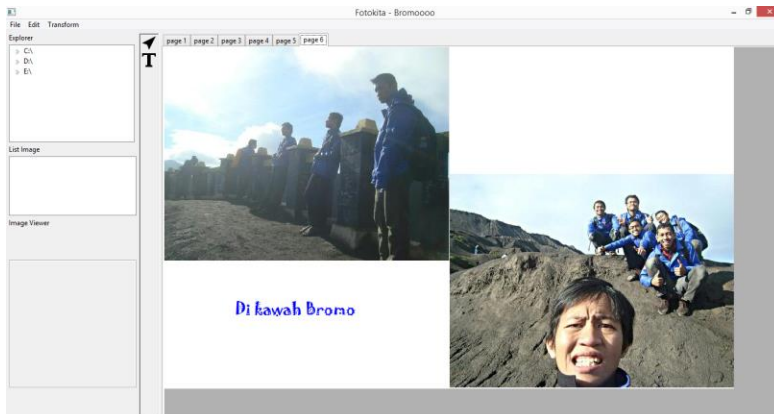
Gambar 5.10 Pemberitahuan Untuk Memperbarui Aplikasi


```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <fotokita version="2">
3      <author>Fadri</author>
4      <createdat>2016-06-21T16:08:13.370+07:00</createdat>
5      <lastmodified>2016-06-21T16:23:04.547+07:00</lastmodified>
6      <pageamount>6</pageamount>
7      <pages size="A4">
8          <page no="0" uuid="ab6ed746-87cf-4d27-9cea-62c8c24865b1">
25         </page>
26         <page no="1" uuid="59edcfbd-caf5-468a-9115-ebb11b58f533">
36         </page>
37         <page no="2" uuid="3794fc3d-ba75-4fbe-b261-f573c0abbf17">
45         </page>
46         <page no="3" uuid="93e24007-baae-4883-9a54-4f972996c2fd">
56         </page>
57         <page no="4" uuid="a06907a0-43e1-4bbb-9aa0-17e596b7ec5e">
64         </page>
65         <page no="5" uuid="af5753b4-f894-413a-9739-a4ce0b8ba468">
79         </page>
80     </pages>
81     <title>Bromoooo</title>
82     <location> Penanjakan Gunung Bromo </location>
83     <uuid>936ca21d-5d88-4d13-b35d-12bc6b66361a</uuid>
84 </fotokita>
85

```

Gambar 5.11 Struktur Metadata File Proyek Versi Baru



Gambar 5.12 File Versi Baru yang dibuka pada Aplikasi Versi Lebih Rendah

5.2.4.2 Pengujian Backward Compatibility

Pengujian ini bertujuan membuktikan aplikasi dengan versi lebih tinggi dapat mengakomodasi *file* proyek yang disimpan dalam versi lebih rendah. Skenario pengujian *backward compatibility* terdapat pada Tabel 5.6. Hasil pengujian dapat dilihat pada Gambar 5.13 dan Gambar 5.14.

```

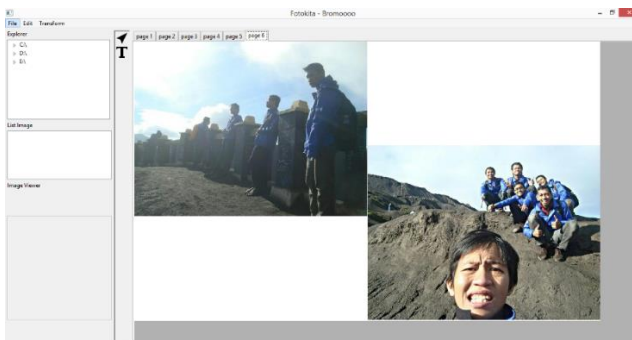
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <fotokita version="1">
3   <author>Fadri</author>
4   <createdat>2016-06-21T16:08:13.370+07:00</createdat>
5   <lastmodified>2016-06-21T16:14:25.564+07:00</lastmodified>
6   <pageamount>6</pageamount>
7   <pages size="A4">
8     <page no="0" uuid="ab6ed746-87cf-4d27-9cea-62c8c24865b1">
25     </page>
26     <page no="1" uuid="59edcfbd-caf5-468a-9115-ebb11b58f533">
36     </page>
37     <page no="2" uuid="3794fc3d-ba75-4fbe-b261-f573c0abbf17">
45     </page>
46     <page no="3" uuid="93e24007-baae-4883-9a54-4f972996c2fd">
56     </page>
57     <page no="4" uuid="a06907a0-43e1-4bbb-9aa0-17e596b7ec5e">
64     </page>
65     <page no="5" uuid="af5753b4-f894-413a-9739-a4ce0b8ba468">
66     <holder uuid="7f583964-c6be-4f25-82ab-d0cc4ea0bcfb">
71     </holder>
72     <holder uuid="c8b45c20-bbd9-4f6c-9f35-93c70a02de35">
77     </holder>
78     </page>
79   </pages>
80   <title>Bromoooo</title>
81   <uuid>936ca21d-5d88-4d13-b35d-12bc6b66361a</uuid>
82 </fotokita>
83

```

Gambar 5.13 Struktur Metadata File Proyek Versi Lebih Rendah

Tabel 5.6 Skenario Uji Coba Backward Compatibility

ID	UJ-005
Kasus Penggunaan	Membaca file proyek dengan versi aplikasi sebelumnya menggunakan aplikasi dengan versi terbaru
Nama	Backward Compatibility
Tujuan Pengujian	Menguji kemampuan sistem untuk mengakomodasi pembaruan yang dilakukan pada sistem
Skenario	Pengguna membuka file proyek versi lama menggunakan aplikasi versi lebih rendah
Kondisi Awal	Pengguna membuka <i>file</i> .fbk dengan aplikasi versi lebih tinggi
Langkah Pengujian	-
Kondisi yang diharapkan	Aplikasi menampilkan proyek ke dalam kanvas dengan mengimplementasi-kan fitur-fitur yang ada pada <i>file</i> proyek
Kondisi yang didapatkan	Aplikasi mengimplementasikan seluruh fitur elemen yang terdapat pada <i>file</i> proyek
Hasil Pengujian	Berhasil



Gambar 5.14 File Versi Lebih Rendah yang dibuka pada Aplikasi Versi Lebih Tinggi

5.3 Skenario Pengujian Konsistensi

Pengujian konsistensi dilakukan terhadap masing-masing skenario pada uji fungsionalitas sebanyak 10 kali. Dilakukan 10 uji fungsionalitas terhadap setiap skenario uji fungsionalitas untuk menguji konsistensi penyimpanan dan pembacaan aplikasi. Hasil pengujian dapat dilihat pada Tabel 5.7.

Tabel 5.7 Hasil Pengujian Konsistensi

No	Pengujian	Berhasil	Gagal	Tingkat keberhasilan (%)
1	UJ-001	10	0	100%
2	UJ-002	10	0	100%
3	UJ-003	10	0	100%
4	UJ-004	10	0	100%
5	UJ-005	10	0	100%

[Halaman ini sengaja dikosongkan

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. *Metadata* XML dapat mengakomodasi penyimpanan file aplikasi Fotokita. Untuk membuat desain *metadata* yang sesuai dengan aplikasi maka diperlukan analisis terhadap fungsional aplikasi.
2. Penggunaan *forward* dan *backward compatibility* sangat penting untuk aplikasi yang bersifat *agile*, agar dalam setiap versi yang berbeda, aplikasi tetap dapat mengeksekusi *file* proyek.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan analisa dan sistem di masa yang akan datang. Saran-saran didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan.

1. Meningkatkan efisiensi pada penyimpanan *partial updating*, jika bisa dalam penyimpanan mengimplementasi algoritma yang lebih optimal agar dalam efisiensi penyimpanan lebih optimal.
2. Menambahkan keamanan pada *file* hasil simpanan, agar *file* hasil penyimpanan tidak dapat diubah dan dimanipulasi oleh pihak lain serta terjaga integritasnya.

3. Menentukan metode untuk menyimpan *file binary* dalam bentuk XML tanpa mengurangi kualitas *file*.

DAFTAR PUSTAKA

- [1] Ullman, Jeffrey D.; Widom, Jennifer, A First Course In Database System 3rd Edition, New Jersey: Pearson Education Inc, 2007.
- [2] w3schools, “XML Path Tutorial,” w3schools, [Online]. Available: http://www.w3schools.com/xsl/xpath_syntax.asp. [Diakses 24 Mei 2016].
- [3] Oracle,
<https://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>,
[Online]. Available:
<https://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>.
[Diakses 22 Maret 2016].
- [4] w3schools, “XML Schema,” [Online]. Available:
http://www.w3schools.com/xml/schema_intro.asp. [Diakses 10 Mei 2016].
- [5] Microsoft Office, “Office Support,” Microsoft, [Online]. Available: <https://support.office.com/en-us/article/Open-a-Word-2010-document-in-an-earlier-version-of-Word-adfe6ca1-7b18-45a2-ba05-cb1b00ad9935?ui=en-US&rs=en-US&ad=US&fromAR=1>. [Diakses 13 Juni 2016].
- [6] Microsoft, “Microsoft MSDN,” Microsoft, 2011. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff649901.aspx>. [Diakses 22 Maret 2015].
- [7] M. A. Ridwan, “Rancang Bangun Aplikasi Editor Album Foto Digital 'Fotokita' berbasis Desktop,” Surabaya, 2016.
- [8] Microsoft MSDN, Microsoft, [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff649901.aspx>. [Diakses 13 Juni 2016].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Fadrian Merdianto, lahir di Jember, 13 Desember 1993. Menempuh pendidikan formal di TK Al-Amien Jember pada tahun 1999-2000, SDN Kebonsari VIII Jember (2000-2006), SMPN 2 Jember (2006-2009), SMAN 1 Jember (2009-2012), dan pendidikan sarjana S1 di Teknik Informatika ITS (2012-2016).

Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTC). Diantaranya adalah menjadi staf Departemen Kewirausahaan dan Minat Bakat, dan Kepala Departemen Pengembangan Profesi HMTC ITS 2014-2015. Penulis juga aktif dalam kegiatan kepanitiaan Schematics dan aktif menjadi Pemandu Latihan Kepemimpinan Manajemen Mahasiswa dari tahun 2013-2015.

Selama kuliah di Teknik Informatika ITS, penulis juga aktif mengikuti beberapa kompetisi, seperti menjadi finalis di Telkomsel NextDev 2015, DevComp FTif Festival, Finalis Hackathon Finhacks BCA 2016, dan juara 2 Indonesia Sociopreneur Challenge 2015 serta Young Sociopreneur di ajang yang sama.