

Rancang Bangun Aplikasi Album Foto Digital 'Fotokita' dengan Penyimpanan Dinamis

Fadrian Merdianto, R.V. Hari Ginardi, dan Rizky Januar Akbar

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: hari@its.ac.id

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—*Backward Compatibility*, *Forward Compatibility*, foto digital, *metadata*

I. PENDAHULUAN

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 markap 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.

II. TINJAUAN PUSTAKA

A. XML

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. 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 [1].

B. XPath

XPath merupakan 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 [2].

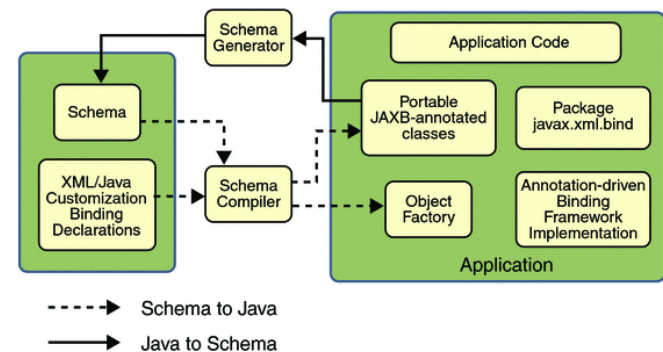
C. Java JAXB 2.0

Java Architecture for XML Binding (JAXB) merupakan 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 dokumen ke dalam Java, dan kemudian menyusun pohon (menulis) konten Java kembali ke dokumen XML. Komponen yang menyusun implementasi JAXB dapat dilihat pada Gambar 1. Keterangan dari Gambar 1 adalah sebagai berikut [3].

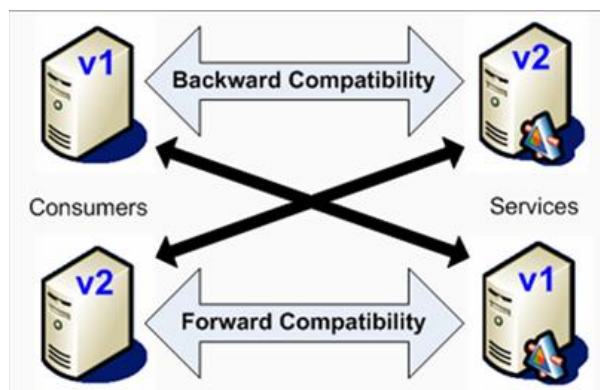
- *Schema compiler*: Membungkus skema sumber ke dalam sebuah elemen skema turunan program.
- *Schema generator*: Memetakan elemen-elemen yang telah

dibuat berdasarkan program ke dalam skema turunan.

- *Binding runtime frameworks*: menyediakan fungsi *unmarshalling* (membaca) dan *marshalling* (menulis) dokumen.



Gambar 1. Arsitektur JAXB



Gambar 2. Skema kerja *Forward Compatibility* dan *Backward Compatibility*

D. 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 berikut [4] [5]:

- 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.

E. Forward Compatibility

Forward Compatibility merupakan 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* aplikasi Microsoft Office Word

2007, namun tidak mengimplementasi beberapa fitur seperti sitasi, persamaan secara optimal.

Tabel 1.

Objek penyusun metadata Fotokita

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
10.	Teks	Teks yang ditambahkan untuk menambah keterangan gambar objek

F. Backward Compatibility

Backward Compatibility merupakan kemampuan sebuah sistem baru untuk menerima, membaca, melihat, dan mengimplementasi fitur yang ada pada sistem versi sebelumnya. 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 2 [5] [6].

III. PERANCANGAN SISTEM

A. Analisis

Sistem yang dibuat merupakan implementasi dari aplikasi berbasis *desktop* Fotokita [7]. Penyimpanan dari aplikasi tersebut disimpan menjadi sebuah *file* objek .fbk yang tersusun dari susunan *metadata*. *File* yang telah dibuat oleh pengguna 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.

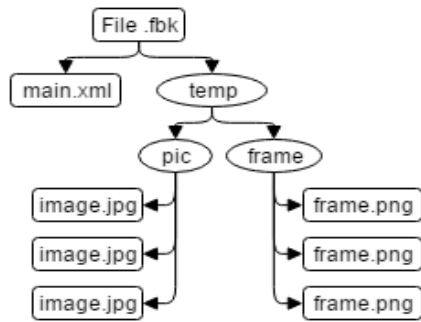
B. Analisis Desain Aplikasi

Berdasarkan analisis terhadap aplikasi *editor* Fotokita, disusun objek yang dibutuhkan agar kebutuhan pada aplikasi *desktop* dan *web* dapat diakomodasi [8]. Setiap *file* yang disusun dalam penyimpanan pada aplikasi ini memenuhi beberapa objek yang dijabarkan pada Tabel 1.

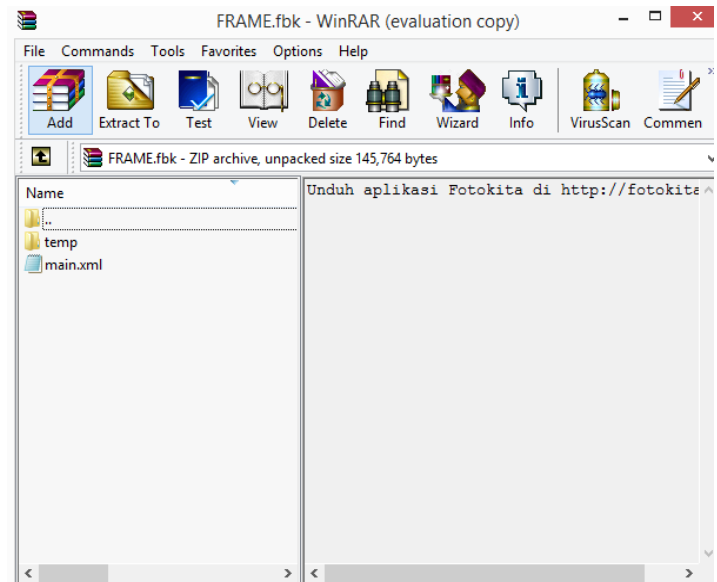
C. Perancangan Struktur File

File yang disimpan oleh aplikasi *editor* dibungkus dalam format ekstensi .fbk. Struktur *file* dapat dilihat pada Gambar 3. 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*

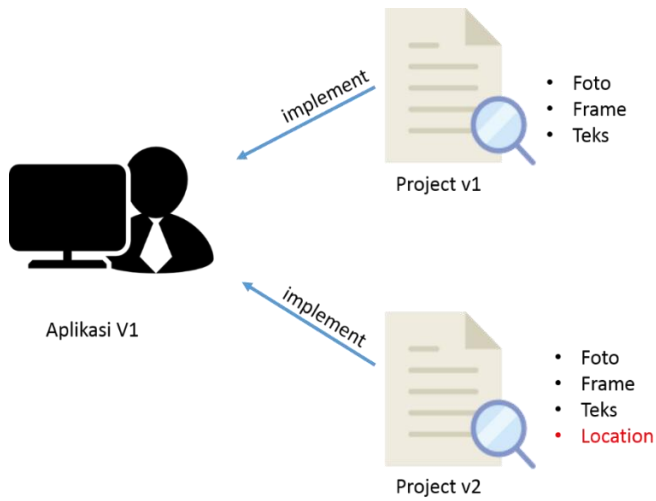
yang nantinya akan langsung dimuat dan diimplementasikan pada *editor* ketika *file* tersebut dibuka oleh aplikasi.



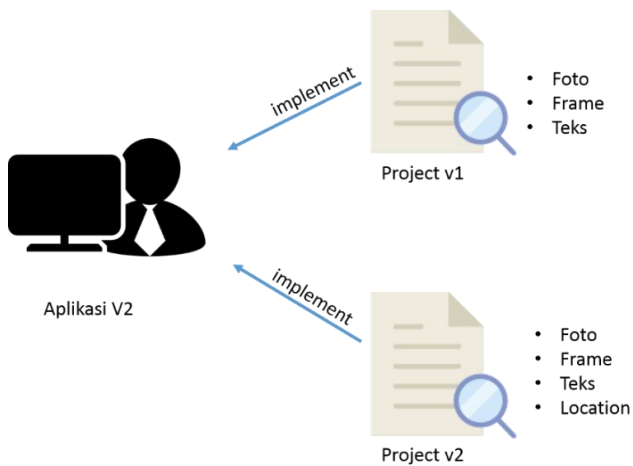
Gambar 3. Struktur *file* .fbk



Gambar 6. Struktur objek penyusun *file* .fbk



Gambar 4. Perancangan skema *Forward Compatibility*



Gambar 5. Perancangan skema *Backward Compatibility*

Tabel 2.
Spesifikasi lingkungan 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

D. Perancangan Skema *Forward Compatibility* dan *Backward Compatibility*

Perancangan skema *forward compatibility* mengakomodasi kemungkinan sebuah aplikasi versi lebih rendah membaca *file* yang disimpan dalam versi lebih tinggi. Karena sistem aplikasi versi lebih rendah tidak memiliki fitur yang sama persis dengan sistem aplikasi versi lebih tinggi, maka diperlukan sebuah skema yang mampu mengantisipasi kesalahan keadaan tersebut. Pada umumnya, kondisi ini diantisipasi dengan tidak mengimplementasi fitur dari *file* versi lebih tinggi yang tidak ada pada sistem aplikasi versi lebih rendah. Keadaan tersebut dapat dilihat pada Gambar 4. Pada Gambar 4, fitur *location* pada versi 2 berwarna merah karena fitur tersebut tidak diimplementasikan pada aplikasi versi 1.

Perancangan skema *backward compatibility* mengakomodasi kemungkinan sebuah aplikasi versi lebih tinggi membaca *file* yang disimpan dalam versi lebih rendah. Pada kasus ini, sistem aplikasi baru memiliki fitur yang lebih banyak dari *file* versi lebih rendah. Karena sistem aplikasi versi lebih tinggi dapat mengakomodasi fitur yang terdapat pada *file* versi lebih rendah, maka sistem aplikasi versi lebih tinggi dapat mengimplementasi seluruh fitur yang terdapat pada *file* versinya maupun lebih rendah, sehingga pada skema *Backward Compatibility* ini tidak ada perubahan terhadap *file* yang diimplementasikan dalam aplikasi. Keadaan tersebut dapat dilihat pada Gambar 5.

Tabel 3.
Daftar pengujian

No	Kode Pengujian	Keterangan
1	UJ-001	Pengujian Menyimpan File
2	UJ-002	Pengujian Membaca File dengan Versi Aplikasi yang sama
3	UJ-003	Pengujian Menyimpan File yang Telah diubah
4	UJ-004	Pengujian <i>Forward Compatibility</i>
5	UJ-005	Pengujian <i>Backward Compatibility</i>

Tabel 4.
Hasil pengujian

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

IV. IMPLEMENTASI

A. Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak. Spesifikasi perangkat keras yang digunakan dalam membangun aplikasi dituliskan pada Tabel 2.

Perangkat lunak yang digunakan untuk membuat aplikasi Fotokita antara lain sebagai berikut:

- *Eclipse Luna*, digunakan sebagai *IDE*.
- *StarUML*, digunakan untuk merancang diagram kelas.

B. Implementasi Struktur File

Struktur *file* yang diimplementasikan sesuai dengan perancangan ada pada Gambar 6. 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.

C. Implementasi Forward dan Backward 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 anak dari elemen *photokita*. Penambahan tersebut memiliki struktur yang sama dengan elemen yang ada pada versi lebih tinggi.

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

V. UJI COBA DAN HASIL

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 ditunjukkan pada Tabel 2. Pengujian dilakukan dengan beberapa skenario yang dijelaskan pada Tabel 3. Pengujian dilakukan sebanyak 10 kali pada masing-masing skenario.

Tabel 4 menunjukkan hasil pengujian, di mana hasil dari seluruh skenario yang dilakukan berhasil sesuai skenario.

VI. 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.
2. Penggunaan *forward* dan *backward compatibility* sangat penting untuk aplikasi yang bersifat *agile*, agar dalam setiap versi yang berbeda, aplikasi tetap dapat mengeksekusi *file*.

DAFTAR PUSTAKA

- [1] Liam, "Extensible Markup Language (XML)," w3c, 19 Mei 2015. [Online]. Available: <http://www.w3.org/XML/>. [Accessed 23 November 2015].
- [2] w3c, "XML Path Language (XPath)," w3c, [Online]. Available: <http://www.w3.org/TR/xpath/>. [Accessed 23 November 2015].
- [3] Oracle, <https://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>, [Online]. Available: <https://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>. [Accessed 22 Maret 2016].
- [4] w3schools, "XML Schema," [Online]. Available: http://www.w3schools.com/xml/schema_intro.asp. [Accessed 10 Mei 2016].
- [5] Ullman, Jeffrey D.; Widom, Jennifer, A First Course In Database System 3rd Edition, New Jersey: Pearson Education Inc, 2007.
- [6] Microsoft, "Microsoft MSDN," Microsoft, 2011. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff649901.aspx>. [Accessed 22 Maret 2015].
- [7] M. A. Ridwan, "Rancang Bangun Aplikasi Editor Album Foto Digital 'Fotokita' berbasis Desktop," Surabaya, 2016.
- [8] D. Siahaan, Analisa Kebutuhan dalam Rekayasa Perangkat Lunak, Yogyakarta: Andi, 2012.