

**Rancang Bangun Sistem Pelacakan Dokumen
Memanfaatkan FUSE dan Samba *File-System*: Studi
Kasus PT. Aneka Tuna Indonesia**

Nama Mahasiswa : I GEDE ADHIARTA WIANDANA
NRP : 5110100171
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Royyana Muslim Ijtihadie, S.Kom.,
M.Kom., Ph.D.
Dosen Pembimbing 2 : Henning Titi Ciptaningtyas, S.Kom.,
M.Kom.

ABSTRAK

Saat ini dokumen digital merupakan aspek penting dalam dunia bisnis. Dokumen digital memuat data-data penting perusahaan baik berupa data produksi sampai data keuangan. Dokumen digital dalam perusahaan dapat diakses oleh orang-orang dalam perusahaan dengan menggunakan media berbagi data. Sehingga dokumen digital rentan terhadap kebocoran baik dari dalam maupun dari luar perusahaan. Dan apabila terjadi kebocoran maka perusahaan akan banyak menderita kerugian. Untuk mencegah terjadinya kebocoran dokumen digital khususnya dari dalam perusahaan, maka perlu pengamanan terhadap dokumen digital dengan cara menyisipkan sebuah informasi khusus ke dalam setiap dokumen digital yang disalin oleh orang-orang dari dalam perusahaan.

Berdasarkan permasalahan diatas, maka dibuatlah suatu sistem yang bisa memasukkan sebuah informasi secara transparan ke dalam dokumen digital yang disalin oleh orang-orang yang berada di dalam perusahaan dengan menggunakan FUSE dan Samba sebagai media berbagi data.

Dari ujicoba yang dilakukan, informasi yang berhasil dimasukkan ke dalam beberapa jenis dokumen digital. Akan tetapi,

mempengaruhi waktu dan kecepatan pengiriman dokumen. Hasil ini menunjukkan bahwa penyisipan informasi secara transparan ke dalam dokumen digital dapat dilakukan dengan menggunakan FUSE walaupun mempengaruhi waktu dan kecepatan pengiriman dokumen digital.

Kata kunci: Dokumen Digital, FUSE, Kebocoran, Penyisipan Informasi, Samba, Transparan.

Design of Document Tracking System Utilizing FUSE and Samba File-System: Case Study in PT. Aneka Tuna Indonesia

Student's Name : I GEDE ADHIARTA WIANDANA
Student's ID : 5110100171
Department : Teknik Informatika FTIF-ITS
First Advisor : Royyana Muslim Ijtihadie, S.Kom.,
M.Kom., Ph.D
Second Advisor : Henning Titi Ciptaningtyas,
S.Kom., M.Kom.

ABSTRACT

Nowadays digital document is an important aspect in the business world. Digital documents contain important data in the form of enterprise production data to financial data. Digital documents of the company can be accessed by people within the company using the media to share data. So that digital documents are prone to be leaked from both inside and outside the company. And if there is a leakage of information then the company will suffer many losses. To prevent leakage of digital documents, especially within the company, it is necessary to make security system for digital documents by inserting a specific information into each digital document copied by people in the company.

Based on the issues above, then company need a system that can insert a information transparently into digital documents copied by the peoples of company by using FUSE and Samba as a medium for sharing data.

From the tests conducted, the information was successfully inserted into several types of digital documents. However, it influences the timing and speed of document delivery. These results

indicate that the insertion of information transparently into digital documents can be done using FUSE although it affects the timing and speed of delivery of digital documents.

Keywords: Digital Documents, FUSE, Leaks, Information Insertion, Samba, Transparent.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1. Dokumen Digital

Dokumen merupakan suatu sarana transformasi informasi dari satu orang ke orang lain atau dari satu kelompok ke kelompok lain. Dokumen meliputi berbagai kegiatan yang diawali dengan bagaimana suatu dokumen dibuat, dikendalikan, diproduksi, disimpan, didistribusikan dan digandakan. Dokumen Digital merupakan setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal atau sejenisnya tetapi tidak terbatas pada tulisan, suara, gambar, peta atau sejenisnya [1]. Dokumen Digital sangat penting dalam kehidupan sehari-hari, organisasi maupun bisnis. Sehingga diperlukan pengamanan terhadap sebuah dokumen digital. Terdapat berbagai macam teknik dalam pengamanan sebuah dokumen digital diantaranya enkripsi, DRM dan masih banyak lainnya. Teknik-teknik penggunaannya berbeda-beda sesuai dengan kebutuhan pengguna. Untuk menangani permasalahan penyebaran dokumen oleh pihak-pihak yang tidak bertanggung jawab di dalam perusahaan, maka perlu disisipkan sebuah informasi tentang siapa dan kapan pegawai menyalin dokumen tersebut. Sehingga apabila terjadi kebocoran dokumen dalam perusahaan dapat dengan mudah diketahui pelaku penyebaran dokumen tersebut.

Dokumen digital yang digunakan dalam penelitian ini adalah dokumen Microsoft Office yang terdiri dari Microsoft Office Word dan Excel, dokumen PDF.

2.2. Linux

Linux merupakan sebuah sistem operasi yang bersifat *open source* dan *free*. Linux memiliki *kernel* yang bernama *linux kernel*. Linux *kernel* bersifat unik dan fleksibel karena dapat diubah sesuai keinginan. Linux dan *linux kernel* dibuat menggunakan banyak bahasa pemrograman yang berbeda antara lain bahasa C, C++, Java, Perl, dan Lisp. Terdapat berbagai macam jenis linux antara lain Ubuntu, Linux Mint, OpenBSD, Solaris dan masih banyak lainnya.

Linux digunakan sebagai sistem operasi pada server sehingga dapat dipasangkan Samba sebagai media berbagi dokumen dan FUSE sebagai sarana untuk membuat *filesystem* untuk menambahkan informasi ke dalam dokumen. Jenis linux yang digunakan pada server adalah Ubuntu versi 12.04.

2.3. Samba

Samba merupakan perangkat lunak yang bersifat *open source/free software* yang menyediakan layanan untuk berbagi data dan servis printer ke pengguna [2]. Karena bersifat *open source*, konfigurasi Samba dapat diubah sesuai dengan kebutuhan. Samba dapat dijalankan pada berbagai macam sistem operasi kecuali Microsoft Windows seperti UNIX, Linux, IBM, OpenVMS dan lain-lain. Samba menggunakan aturan SMB/CIFS untuk melakukan pengiriman data dari server menuju pengguna dan sebaliknya. Samba memungkinkan untuk melakukan integrasi antara server yang memiliki sistem operasi Linux/Unix dengan pengguna yang memiliki sistem operasi Microsoft Windows untuk melakukan pertukaran data dan dokumen. Konfigurasi Samba dilakukan pada data bernama *smb.conf*. Konfigurasi yang dapat dilakukan pada Samba antara lain konfigurasi keamanan, konfigurasi riwayat penggunaan, konfigurasi akses direktori, konfigurasi IP dan *port*, konfigurasi basis data dan lain-lain. Gambar 2.1 menunjukkan contoh konfigurasi Samba pada *smb.conf*.

```
[global]
    workgroup = SIMPLE
[test]
    comment = For testing only, please
    path = /export/samba/test
    read only = no
    guest ok = yes
```

Gambar 2.1 Contoh konfigurasi Samba pada *smb.conf*

Dalam sistem yang akan dibangun, Samba digunakan sebagai media layanan berbagi data dan dokumen yang dibagikan oleh server. Selain itu Samba akan dikonfigurasi sehingga dapat mencatat aktivitas pengambilan dokumen dari *File Server* menuju *File Copy Station* dan dikonfigurasi agar memiliki basis data untuk menampung pengguna pegawai. Basis data yang dikonfigurasi terbatas hanya mampu menampung 250 pengguna pegawai. Versi Samba yang digunakan dalam pembuatan sistem adalah Samba 3.6.3.

2.4. Syslogd

Syslogd merupakan sebuah perangkat lunak yang berjalan pada sistem operasi linux yang berfungsi untuk mencatat riwayat sistem dan kernel. Selain itu syslogd juga dapat mencatat riwayat dari program-program yang berjalan secara otomatis.

Dalam sistem yang dibangun syslogd digunakan untuk melakukan pencatatan aktivitas riwayat samba sesuai dengan konfigurasi yang dipasang pada samba. Riwayat yang dicatat adalah username, waktu akses, IP pengguna, nama mesin yang digunakan untuk mengakses dan operasi yang dilakukan.

2.5. Hash Function

Hash Function merupakan sebuah fungsi yang berguna untuk mengkompresi/memperkecil sebuah *string* yang panjang menjadi lebih pendek. Terdapat bermacam-macam *hash function* antara lain sha1, md4, md5 dan masih banyak yang lainnya [3].

Dalam sistem *hash function* digunakan untuk menjaga integritas dokumen. Dan *hash function* yang digunakan dalam pembuatan sistem adalah *md5*.

2.6. FUSE

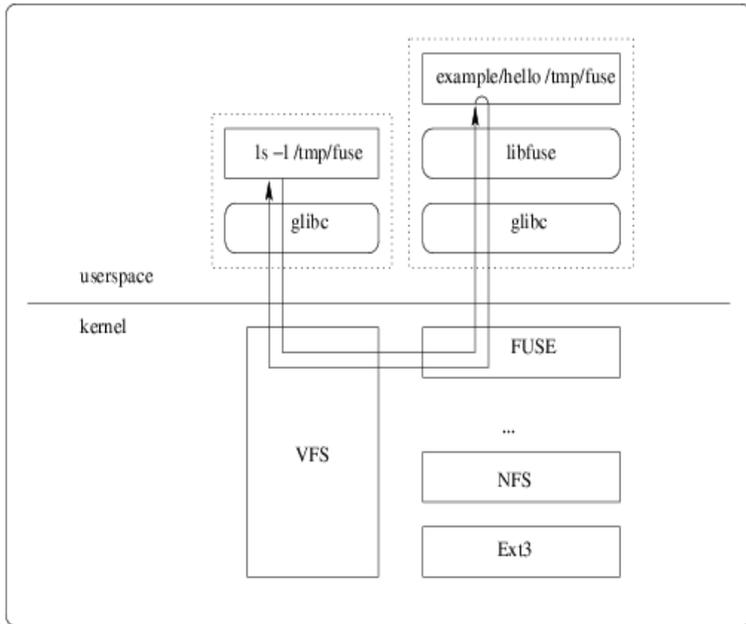
FUSE yang merupakan kependekan dari *Filesystem in Userspace* merupakan sebuah mekanisme pada sistem operasi yang memungkinkan pengguna untuk membuat sebuah *filesystem* sendiri tanpa mengubah kode pada *kernel* [4]. Modul FUSE hanya menyediakan jembatan antarmuka kernel. Modul FUSE dapat dijalankan pada sistem operasi kecuali Microsoft Windows antara lain Linux, Unix, MAC OS dan lain-lain. Gambar 2.2 menunjukkan diagram alur bagaimana FUSE bekerja.

FUSE dalam sistem akan digunakan sebagai sarana untuk membuat sebuah *filesystem* dimana *filesystem* tersebut nantinya akan digunakan untuk menyisipkan informasi ke dalam data dan dokumen yang disalin oleh pegawai dari media berbagi data yang dibagikan oleh server menggunakan Samba sebagai medianya.

2.6.1. Fusepy

Fusepy merupakan sebuah modul Python yang menyediakan sebuah antarmuka yang sederhana untuk implementasi FUSE dan MacFUSE. *Fusepy* ini hanya berupa sebuah data dan dibuat menggunakan *ctypes*. *Fusepy* dapat mengeksekusi perintah-perintah pada FUSE seperti *open*, *write*, *read*, *getattr*, *readdir*, *mkdir*, *chmod*, *chown* dan masih banyak lainnya.

Fusepy pada sistem digunakan sebagai pustaka untuk membuat sebuah *filesystem* menggunakan FUSE dengan menggunakan bahasa pemrograman Python. Versi pustaka *fusepy* yang digunakan dalam pembuatan sistem adalah *fusepy 2.0.2*.



Gambar 2.2 Diagram Alur FUSE bekerja [4]

2.7. Python

Python merupakan salah satu bahasa pemrograman yang dapat digunakan untuk keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi diantaranya adalah Linux/Unix, Microsoft Windows, Mac OS dan lain-lain. Python mendukung berbagai macam modul dan paket yang digunakan untuk melakukan pengembangan terhadap perangkat lunak.

Python pada sistem yang dibuat merupakan bahasa pemrograman yang digunakan untuk membuat sebuah *filesystem* menggunakan FUSE dengan pustaka *fusepy*. Selain itu digunakan untuk membuat sebuah fungsi menambahkan informasi ke dalam data dan dokumen sebelum data dan dokumen tersebut disalin oleh pegawai. Versi python yang digunakan dalam pembuatan sistem adalah python 2.7

2.8. XML

XML yang merupakan kependekkan dari *Extensible Markup Language* adalah sebuah dokumen yang tersusun dari beberapa *tags* yang mendeskripsikan sebuah struktur. Tiap *tags* pada XML berisikan konten data. Sebuah data XML terdiri dari atribut, komentar, proses instruksi, objek referensi, *namespace* dan data. Gambar 2.3 menunjukkan contoh sebuah data XML.

Teknologi XML digunakan pada Microsoft Office versi 2007 dan 2010, dimana xml digunakan untuk menyimpan informasi-informasi tentang dokumen berupa dokumen properti, isi dokumen, tema dokumen dan lain-lain.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <title>
    My First XML Document
  </title>
  <para>
    Hello to the wild and wooly world of electronic
    document encoding.
  </para>
  <note>
    (Okay, so maybe it's not that wild,
    but it can get pretty wooly.)
  </note>
  <!-- No simple "Hello World" here -->
  <para>
    This page written by:
    <ital>(insert your name here)</ital>
  </para>
  <para>
    Copyright 2000 and beyond
  </para>
</root>
```

Gambar 2.3 Contoh data XML

2.9. Microsoft Office

Microsoft Office merupakan kumpulan perangkat lunak yang didesain untuk membantu membuat sebuah dokumen. Tiap perangkat lunak pada Microsoft Office memiliki kegunaan yang berbeda dan beragam sesuai dengan kebutuhan pengguna. Adapun perangkat lunak yang termasuk ke dalam kumpulan perangkat lunak pada Microsoft Office adalah sebagai berikut:

1. Microsoft Office Word,
2. Microsoft Office Excel,
3. Microsoft Office PowerPoint,
4. Microsoft Office Publisher,
5. Microsoft Office Access

4 versi terakhir yang telah dikeluarkan oleh Microsoft Office antara lain Microsoft Office 2003, Microsoft Office 2007, Microsoft Office 2010 dan Microsoft Office 2013. Pada penelitian ini digunakan versi Microsoft Office 2007 dan Microsoft Office 2010 karena pada kedua versi ini menggunakan bentuk Office Open XML.

2.9.1. Microsoft Office XML

Pada versi Microsoft Office 2007 dan 2010, perangkat lunak Microsoft Office menggunakan bentuk Office Open XML dengan teknologi ZIP dengan tujuan untuk mengurangi ukuran dokumen, mempermudah melakukan pengecekan masalah dan untuk mempermudah melakukan proteksi terhadap dokumen yang mengandung informasi yang sensitif. Office Open XML menyimpan berbagai informasi *document body*, *headers* dan *comments*. Gambar 2.4 menunjukkan struktur dari sebuah dokumen Microsoft Office.

Name ▲	Size	Packed	Type	Modified	CRC32
..			Folder		
_rels			Folder	6/11/2014 1:22 PM	
docProps			Folder	6/11/2014 1:22 PM	
word			Folder	6/11/2014 1:22 PM	
[Content_Types].xml	987	987	XML Document	6/11/2014 1:22 PM	606AA6AA

Gambar 2.4 Struktur sebuah dokumen Microsoft Office versi 2007

Masing-masing dari informasi tersebut disimpan ke dalam sebuah XML yang berbeda. Pada Office Open XML yang telah dikompresi oleh ZIP terdapat 4 buah data yaitu direktori *_rels* yang

menyimpan informasi tentang hubungan antar XML dalam data yang telah dikompresi, direktori *word* yang menyimpan informasi tentang konten dokumen seperti gambar, tulisan dan elemen-elemen dokumen seperti *headers*, *theme*. Direktori *docProps* yang menyimpan informasi tentang dokumen properti seperti nama penulis, nama perusahaan, tanggal pembuatan dokumen dan lain-lain. Dan yang terakhir adalah [*Content_Types*].*xml* yang digunakan untuk mendefinisikan tipe dari konten yang ada dalam file yang telah dikompresi [5].

Untuk memasukan sebuah informasi secara transparan ke dalam dokumen Microsoft Office versi 2007 dan 2010 diperlukan *core.xml* yang berada pada direktori *docProps* untuk ditambahkan informasi. Dalam *core.xml* terdapat properti dokumen antara lain judul dokumen, subjek dokumen, pembuat dokumen, kata kunci, deskripsi dokumen, pengguna terakhir yang melakukan perubahan dokumen, revisi, waktu pembuatan dokumen, waktu perubahan dokumen, kategori dokumen dan konten dokumen. Properti dokumen tersebutlah yang akan ditambahkan informasi secara transparan. Gambar 2.5 menunjukkan struktur dari *core.xml*

```
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dcmitype="http://purl.org/dc/dcmitype/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dc:title> </dc:title>
  <dc:subject> </dc:subject>
  <dc:creator> </dc:creator>
  <cp:keywords> </cp:keywords>
  <dc:description> </dc:description>
  <cp:lastModifiedBy> </cp:lastModifiedBy>
  <cp:revision> </cp:revision>
  <dcterms:created xsi:type="dcterms:W3CDTF"> </dcterms:created>
  <dcterms:modified xsi:type="dcterms:W3CDTF"> </dcterms:modified>
  <cp:category> </cp:category>
  <cp:contentStatus> </cp:contentStatus>
</cp:coreProperties>
```

Gambar 2.5 Struktur *core.xml*

2.9.2. ZipFile

ZipFile merupakan sebuah pustaka yang ada pada bahasa pemrograman Python yang menyediakan fungsi membuat, membaca, menulis, menggabungkan dan memuat daftar data yang ada pada sebuah data ZIP. Pustaka ZipFile menyediakan dekripsi dari data yang terenkripsi dalam data ZIP namun tidak dapat membuat data yang terenkripsi.

Pustaka ZipFile digunakan untuk melakukan ekstraksi data dokumen Microsoft Office. Hasil ekstraksi yang berupa data *core.xml* pada direktori *docProps* pada dokumen Microsoft Office nantinya akan dimasukkan informasi tentang siapa pegawai yang menyalin dokumen dan kapan pegawai tersebut menyalin data tersebut. Selain digunakan untuk ekstraksi dokumen Microsoft Office, pustaka ZipFile juga digunakan untuk membentuk kembali dokumen Microsoft Office yang *core.xml*-nya telah ditambahkan informasi.

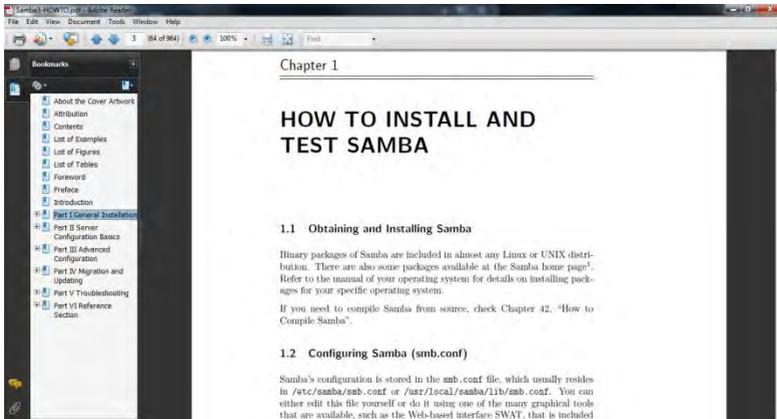
2.9.3. Lxml

Lxml merupakan sebuah modul pustaka yang tersedia untuk bahasa pemrograman Python yang digunakan sebagai alat untuk memproses data XML [6]. Lxml dapat digunakan untuk membuat, membaca, mengubah sebuah data XML.

Pustaka lxml digunakan untuk memasukkan informasi ke dalam *core.xml* yang telah diekstraksi menggunakan pustaka ZipFile dari dokumen Microsoft Office. Versi dari pustaka lxml yang dipakai dalam pembuatan sistem adalah lxml 3.3.5.

2.10. PyPDF

PDF kependekkan dari *Portable Document Format* merupakan sebuah standar untuk pertukaran dokumen elektronik yang berada dalam pengawasan ISO. Ketika melakukan konversi dari dokumen, formulir, halaman web ke PDF, hasilnya terlihat seperti hasil yang dicetak akan tetapi PDF memiliki kelebihan yaitu terdapat tautan dan tombol yang dapat diklik pada dokumen sehingga memudahkan dalam pencarian kata atau kalimat dalam dokumen. Gambar 2.6 menunjukkan contoh dari dokumen PDF.



Gambar 2.6 Contoh dokumen PDF

Untuk memasukkan sebuah informasi ke dalam dokumen PDF dilakukan dengan cara menambahkan informasi tersebut ke dalam *metadata* dokumen PDF tersebut. PyPDF merupakan sebuah modul pustaka yang tersedia untuk bahasa pemrograman Python yang digunakan sebagai alat untuk memproses dokumen PDF. PyPDF dapat digunakan untuk membuat, membaca, mengubah sebuah dokumen PDF.

Pustaka PyPDF digunakan untuk memasukkan informasi ke dalam *metadata* dokumen PDF. Informasi yang dimasukkan ke dalam dokumen PDF sama dengan informasi yang dimasukkan ke dalam *core.xml* yang terdapat dalam dokumen Microsoft Office. Versi dari pustaka PyPDF yang dipakai dalam pembuatan sistem adalah PyPDF 1.13.

2.11. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL (DBMS)* yang *multithread*, dan *multi-user*. *MySQL* adalah implementasi dari sistem manajemen basis data relasional (*RDBMS*). *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses data yang terbagi menjadi tiga bagian, yaitu *DDL*,

DML, dan *DCL*. Berikut ini beberapa kelebihan *MySQL* sebagai database *server* antara lain:

1. *Source MySQL* dapat diperoleh dengan mudah dan gratis.
2. *MySQL* merupakan program yang *multithread*, sehingga dapat dipasang pada *server* yang memiliki *multiCPU*.
3. Didukung program-program umum seperti *C*, *C++*, *Java*, *Perl*, *PHP*, *Python*, dsb.
4. Bekerja pada berbagai *platform*.

Implementasi *MySQL* pada sistem yang dibuat adalah sebagai tempat penyimpanan riwayat informasi penyalinan dokumen.

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Perancangan merupakan bagian penting dari pembuatan perangkat lunak yang isinya berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses, alur dan implementasinya.

3.1. Deskripsi Umum

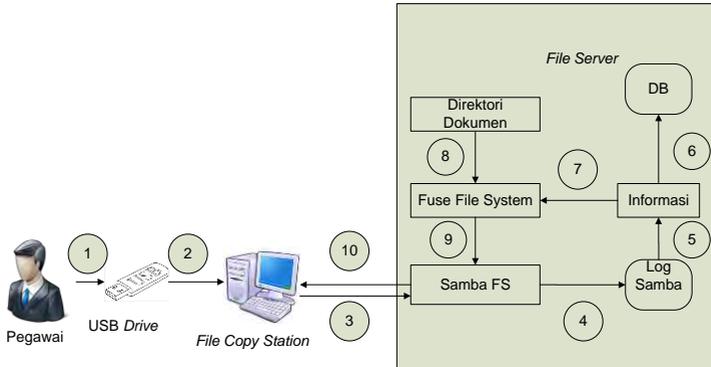
Pada tugas akhir ini akan dikembangkan suatu sistem keamanan dokumen pada server yang berguna sebagai sarana untuk memasukkan informasi yang berupa *username* pegawai dan waktu pegawai menyalin dokumen secara transparan ke dalam dokumen yang disalin oleh pegawai dari media berbagi dokumen Samba yang ada di server. Sistem ini nantinya akan diimplementasikan menggunakan FUSE sebagai *filesystem* yang secara otomatis menambahkan informasi ke dalam dokumen yang disalin oleh pegawai.

Gambaran umum dari sistem ini nantinya yaitu, Pertama pegawai menggunakan USB *Drive* untuk mengambil dokumen pada *File Server* dari *File Copy Station*. Sebelum bisa mengambil dokumen melalui *File Copy Station*, pegawai diharuskan *login* pada *File Copy Station*. Setelah berhasil, pegawai dapat menyalin dokumen pada *File Server* melalui *File Copy Station*. Di sisi *File Server*, dokumen yang diminta oleh pegawai disisipkan sebuah informasi secara transparan tanpa diketahui oleh pegawai tersebut sebelum dikirim ke *File Copy Station* melalui FUSE *filesystem*. Informasi yang disisipkan tersebut didapat dari *log* dari Samba. Hal ini dilakukan agar dapat diketahui siapa dan kapan dokumen tersebut disalin apabila terjadi penyalahgunaan. Informasi yang disisipkan ke dalam dokumen disimpan pada *database* di server sebagai riwayat penyalinan dokumen. Kemudian dokumen yang telah disisipi informasi tadi dikirim melalui Samba *File-System* ke

File Copy Station. Lalu dokumen yang telah berisi informasi tersebut disalin menggunakan *USB Drive* oleh pegawai tersebut.

3.2. Arsitektur Umum Sistem

Rancangan arsitektur dari sistem yang dibuat dapat dilihat pada Gambar 3.1.



Gambar 3.1 Arsitektur umum sistem

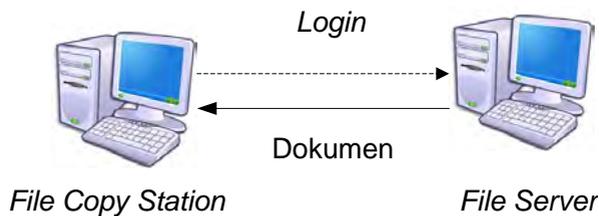
Penjelasan arsitektur umum sistem :

1. Pegawai menggunakan *USB Drive* untuk menyalin data dari *File Server* melalui *File Copy Station*.
2. *USB Drive* ditancapkan ke *File Copy Station*
3. Pegawai melakukan *login* pada *File Copy Station* agar bisa mengakses data pada *File Server* melalui Samba
4. Setelah berhasil melakukan *login*, riwayat *login* tersebut disimpan dalam data riwayat Samba. Lalu pegawai dapat memilih dokumen yang ingin disalin ke dalam *USB Drive*.
5. Seluruh aktivitas pegawai yang tercatat pada riwayat Samba, diambil *username* pegawai sesuai dengan waktu penyalinan dokumen.
6. Informasi yang terdiri dari *username*, waktu penyalinan dan hash dokumen disimpan ke dalam DB sebagai riwayat penyalinan dokumen oleh pegawai pada server.

7. Informasi yang terdiri dari *username*, waktu penyalinan dan hash dokumen akan ditambahkan ke dalam dokumen menggunakan *filesystem*.
8. Dokumen yang akan disalin oleh pegawai akan mengalami penambahan informasi.
9. Dokumen yang telah ditambahkan informasi lalu akan dikirimkan menuju *File Copy Server* melalui Samba.
10. Setelah sampai, dokumen yang telah ditambahkan informasi tersebut disalin ke dalam *USB Drive* oleh pegawai.

3.3. Arsitektur Jaringan

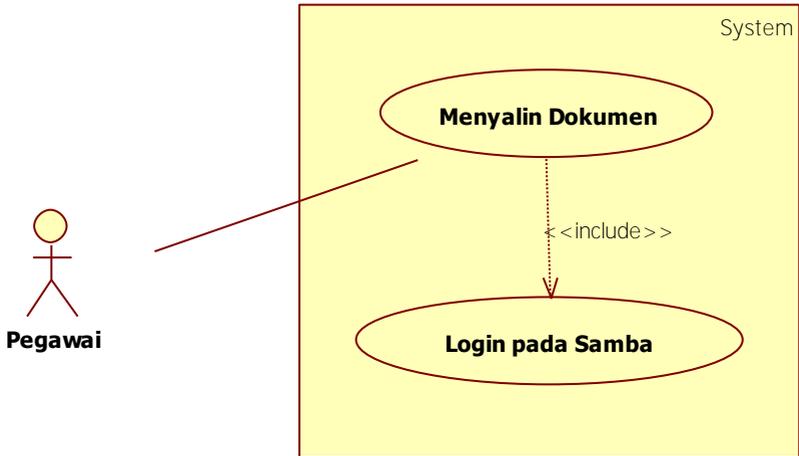
Tugas akhir ini dibangun dengan arsitektur jaringan *client-server* seperti pada Gambar 3.2. Arsitektur jaringan ini disesuaikan dengan keadaan di perusahaan. *Client* di sini berupa *File Copy Station* sebagai tempat untuk menyalin data yang dibagikan oleh media berbagi dokumen Samba pada *File Server*. Dan server di sini berupa *File Server* sebagai tempat menyimpan dokumen yang akan dibagikan ke *File Copy Station* dan sebagai tempat pengimplementasian *filesystem* untuk memasukkan informasi ke dalam dokumen yang disalin oleh pegawai.



Gambar 3.2 Rancangan umum penyalinan dokumen

3.4. Perancangan Diagram Use Case

Diagram *Use Case* menggambarkan fungsionalitas sistem beserta aktor yang terlibat, seperti ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram use case sistem

Pada sistem yang akan dibangun, akan terdapat satu aktor yaitu pegawai yang bertugas menjalankan sistem. Pegawai dapat menjalankan fitur-fitur yang ada pada sistem ini, yaitu:

- **Menyalin Dokumen**
Pegawai dapat menentukan dan menyalin dokumen apa saja yang terdapat dalam media berbagi dokumen yang ada pada *File Server*.
- **Login pada Samba**
Sebelum bisa menentukan dan menyalin dokumen, pegawai terlebih dahulu diharuskan *login* Samba. Setelah berhasil melakukan *login*, maka pegawai berhak melakukan penyalinan dokumen yang terdapat pada media berbagi dokumen pada *File Server*.

3.5. Perancangan Proses

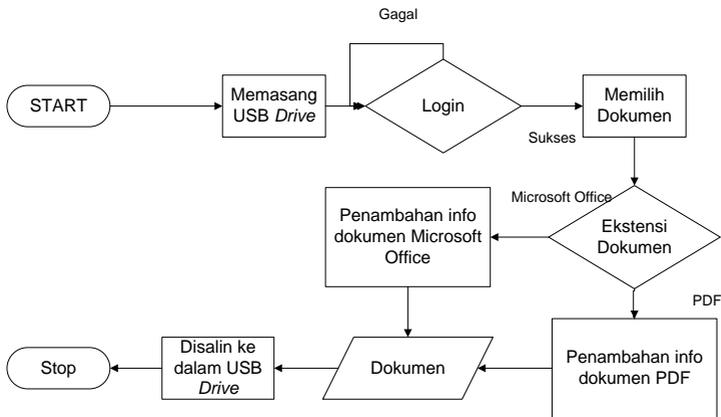
Alur setiap proses yang terdapat pada sistem digambarkan pada diagram alir untuk memudahkan pemahaman secara garis besar proses yang ada di sistem. diagram alir sistem adalah sebagai berikut:

1. Diagram alir penyalinan dokumen
 - a. Diagram alir penyalinan dokumen.
 - b. Diagram alir menambahkan informasi pada dokumen Microsoft Office.
 - c. Diagram alir menambahkan informasi pada dokumen PDF.
 - d. Diagram alir *hash* dokumen dengan menggunakan *md5*.

3.5.1.1. Diagram alir penyalinan dokumen

Proses penyalinan dokumen, seperti pada Gambar 3.4, berjalan pada saat pegawai memasukkan *USB Drive* pada *File Copy Station*.

Pegawai memasang *USB Drive* pada *File Copy Station* untuk menyalin dokumen pada *File Server*. Lalu pegawai diharuskan untuk melakukan *login* terlebih dahulu sebelum bisa menyalin dokumen pada *File Server*. Setelah berhasil melakukan *login* pada *File Copy Station*, pegawai dapat menyalin dokumen yang telah dibagikan oleh media berbagi dokumen yang dibagikan menggunakan Samba pada *File Server*. Sebelum dokumen yang dapat disalin oleh pegawai ke *USB Drive*, akan ditambahkan informasi melalui *filesystem* yang terdapat pada *File Server*. Penambahan informasi tersebut dibedakan menjadi dua cara yaitu penambahan informasi pada dokumen Microsoft Office dan penambahan informasi pada dokumen PDF. Setelah melalui salah satu cara penambahan informasi tersebut, dokumen yang berisi informasi tersebut disalin ke dalam *USB Drive*.

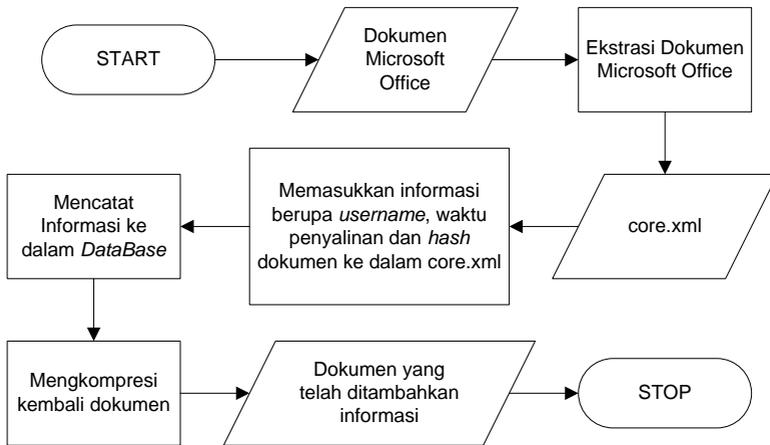


Gambar 3.4 Diagram alir penyalinan dokumen

3.5.1.2. Diagram Alir Penambahan Informasi Pada Dokumen Microsoft Office

Proses penambahan informasi pada dokumen Microsoft Office, seperti pada Gambar 3.5, berjalan pada saat pegawai memilih dokumen Microsoft Office yang akan disalin ke dalam USB Drive.

Dokumen Microsoft Office yang dipilih oleh pegawai untuk disalin akan mengalami proses ekstraksi untuk mendapatkan *core.xml* yang terdapat pada direktori *docProps* pada dokumen Microsoft Office. *Core.xml* ini merupakan XML yang menampung data properti dari dokumen Microsoft Office. Lalu *core.xml* tersebut dimasukkan informasi yang berupa *username* pegawai yang didapatkan dari data riwayat Samba pada bagian penulis dokumen dan waktu pegawai menyalin data pada bagian orang terakhir yang memodifikasi dokumen, serta *hash* dokumen tersebut menggunakan *md5* pada bagian komentar. Kemudian, *core.xml* yang telah ditambahkan informasi tersebut dikompresi bersama data-data dari dokumen Microsoft Office yang telah diekstraksi sebelumnya. Setelah itu dokumen Microsoft Office yang telah berisi informasi tambahan tersebut siap dikirimkan dari *File Server* menggunakan Samba ke *File Copy Station*.

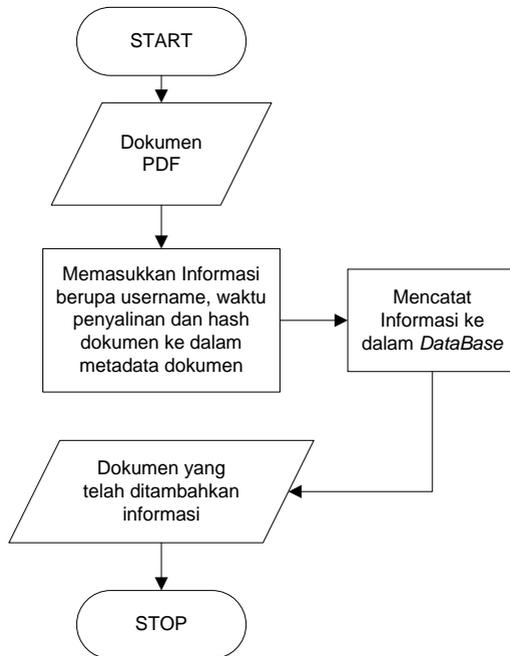


Gambar 3.5 Diagram alir penambahan informasi pada dokumen Microsoft Office

3.5.1.3. Diagram Alir Penambahan Informasi Pada Dokumen PDF

Proses penambahan informasi pada dokumen PDF, seperti pada Gambar 3.6, berjalan pada saat pegawai memilih dokumen PDF yang akan disalin ke dalam *USB Drive*.

Dokumen PDF yang akan disalin oleh pegawai ke dalam *USB Drive* pertama-tama akan ditambahkan informasi dengan cara penambahan informasi pada dokumen PDF. Informasi yang berupa *username* pegawai yang diambil dari data riwayat Samba dan waktu penyalinan serta *hash* dokumen tersebut menggunakan *md5* dimasukkan ke dalam *metadata* dokumen PDF tersebut. Setelah dimasukkan informasi, dokumen PDF tersebut lalu ditutup oleh *filesystem*. Kemudian dokumen PDF tersebut disalin ke dalam *USB Drive* oleh pegawai dari *File Server* melalui *File Copy Station*.

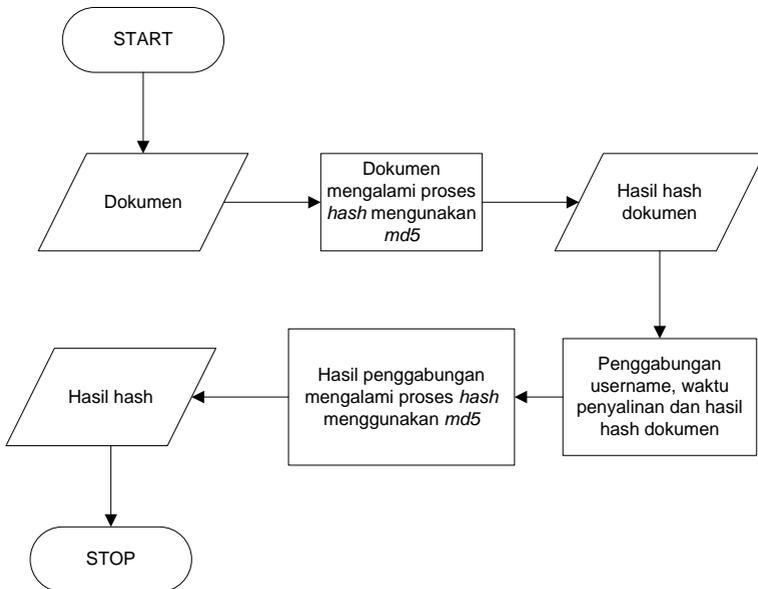


Gambar 3.6 Diagram alir penambahan informasi pada dokumen PDF

3.5.1.4. Diagram Alir Hash Dokumen Menggunakan md5

Proses *hash* dokumen menggunakan md5 dapat dilihat seperti pada Gambar 3.7, berjalan pada sebelum informasi ditambahkan ke dalam dokumen.

Sebelum informasi dimasukkan ke dalam dokumen yang akan disalin, dokumen tersebut mengalami proses *hash* dengan menggunakan algoritma *md5*. Hasil dari proses *hash* tersebut digabungkan dengan *username* dan waktu penyalinan dokumen. Setelah itu, hasil dari penggabungan tersebut mengalami proses *hash* yang nantinya hasil dari proses *hash* ini dimasukkan ke dalam dokumen yang akan disalin oleh pegawai.



Gambar 3.7 Diagram alir *hash* dokumen menggunakan *md5*

3.6 Perancangan Basis Data

Entitas *log_user* pada Basis Data di server digunakan untuk menyimpan riwayat informasi penyalinan dokumen yang dilakukan oleh pegawai. Entitas ini menyimpan *username*, nama dokumen, waktu penyalinan dan hash dari dokumen. Tabel 3.1 menunjukkan detail dari entitas *log_user*.

Tabel 3.1 Properti dari Entitas *log_user*

Nama Properti	Tipe Data	Keterangan
Username	varchar	Username Pegawai
Dokumen	varchar	Nama Dokumen
Waktu	varchar	Waktu Penyalinan Dokumen
Hash	varchar	Hash dokumen

(Halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Bab ini merupakan bahasan mengenai implementasi dari perancangan perangkat lunak yang telah dijabarkan pada bab sebelumnya.

4.1. Lingkungan Pembangunan Perangkat Lunak

Pembangunan perangkat lunak dilakukan pada lingkungan pengembangan sebagai berikut:

4.1.1. Lingkungan Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk pengembangan sistem adalah sebagai berikut:

- Sistem Operasi berupa Windows 7 Ultimate 64 bit pada *client* dan Sistem Operasi berupa Ubuntu 12.04 64 bit pada *server*,
- Microsoft Visio 2007 untuk melakukan perancangan *flowchart* antarmuka aplikasi,
- Star UML untuk mendesain diagram *usecase*,
- Virtual Box untuk pengembangan sistem dan sebagai sarana untuk membuat server,
- Samba sebagai sarana media berbagi dokumen yang dipasang pada server di dalam Virtual Box,
- pustaka fusepy digunakan untuk membuat *filesystem* berbasis FUSE,
- pustaka lxml digunakan untuk mengolah data XML,
- pustaka pyPdf digunakan untuk mengolah dokumen PDF,
- pustaka ZipFile digunakan untuk mengekstraksi dokumen Microsoft Office
- MySQL digunakan sebagai basis data untuk penyimpanan riwayat penyalinan dokumen di server.

4.1.2. Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan sebagai lingkungan pengembangan perangkat lunak tugas akhir adalah sebagai berikut:

- laptop Toshiba dengan *processor* Intel Core i3 M350 2.27GHz,
- memori 4 GB DDR3 dan,
- USB *Drive* Toshiba 16 Gb.

4.2. Implementasi Perangkat Lunak

Implementasi pada perangkat lunak terbagi menjadi lima bagian, yaitu konfigurasi Samba, implementasi filesystem, implementasi penambahan informasi ke dalam dokumen Microsoft Office dan implementasi penambahan informasi ke dalam dokumen PDF. Setiap bagian akan dijelaskan lebih lanjut pada masing-masing subbab.

4.2.1. Konfigurasi Samba

Samba merupakan media berbagi dokumen yang dipasang di sisi server. Samba digunakan untuk membagikan direktori yang berisi dokumen-dokumen yang dibagikan oleh server yang sebelumnya telah dipasang *filesystem*. Konfigurasi yang ditambahkan pada Samba dapat dilihat pada Gambar 4.1.

```
[share]
path = /srv/samba/share/
browseable = yes
read only = no
guest ok = no
valid user = @users
force group = users
vfs object = full_audit
full_audit:prefix = %u|%I|%m|%S
full_audit:success = pread open
full_audit:failure = none
full_audit:facility = local7
full_audit:priority = notice
```

Gambar 4.1 Konfigurasi Samba

Adapun penjelasan dari masing-masing konfigurasi adalah sebagai berikut :

1. *Path* merupakan lokasi direktori yang akan dibagikan oleh Samba pada media berbagi dokumen. *Path* yang diset pada implementasi adalah */srv/samba/share/*.
2. *Browseable* merupakan sebuah konfigurasi di mana media berbagi dokumen dapat diakses atau tidak oleh pengguna. *Browseable* yang diset pada implementasi adalah *yes*. Diset *yes* agar pengguna dapat mengakses media berbagi dokumen.
3. *Read Only* merupakan sebuah konfigurasi di mana media berbagi dokumen hanya memberikan akses untuk membaca sebuah dokumen. *Read Only* yang diset pada implementasi adalah *no*.
4. *Guest Ok* merupakan sebuah konfigurasi untuk menentukan apakah pengguna yang tidak memiliki hak untuk akses media berbagi dokumen untuk mengakses media berbagi dokumen. *Guest Ok* yang diset pada implementasi adalah *no*. Diset *no* agar pengguna yang tidak memiliki hak akses tidak dapat mengakses media berbagi dokumen.
5. *Valid user* merupakan sebuah konfigurasi untuk menentukan siapa saja yang berhak untuk mengakses media berbagi dokumen. *Valid User* pada implementasi diset *@user*, yang berarti user yang tergabung dalam grup user saja yang bisa mengakses media berbagi dokumen.
6. *Force group* merupakan sebuah konfigurasi untuk menentukan grup apa saja yang bisa mengakses media berbagi dokumen. Grup yang bisa mengakses media berbagi dokumen di sini adalah user.
7. *Vfs Object* merupakan sebuah modul pada Samba yang memungkinkan untuk mengakses *filesystem* UNIX. *Vfs Object* yang dipakai di sini adalah *full_audit*. *Full_audit* di sini memungkinkan untuk mencatat riwayat akses pengguna ke media berbagi dokumen Samba secara otomatis dengan menggunakan bantuan *syslog* [7]. *Syslog* bekerja secara otomatis pada Linux untuk mencatat riwayat aplikasi yang

sedang berjalan. Adapun konfigurasi *full_audit* yang diimplementasikan pada Samba antara lain:

- ***Full_audit:prefix*** merupakan konfigurasi untuk mencatat *prefix* pada riwayat Samba. *Prefix* yang ditampilkan pada riwayat Samba antara lain %u untuk menampilkan username, %I untuk menampilkan alamat IP pengguna, %m untuk menampilkan *username* mesin yang dipakai untuk mengakses media berbagi dokumen oleh pengguna, %S untuk menampilkan dokumen apa yang diakses oleh pengguna pada media berbagi dokumen.
- ***Full_audit:success*** merupakan konfigurasi untuk mencatat operasi apa saja yang berhasil dilakukan oleh pengguna pada media berbagi dokumen. Operasi yang dicatat pada riwayat Samba adalah *pread* dan *open*.
- ***Full_audit:failure*** merupakan konfigurasi untuk mencatat operasi apa saja yang gagal dilakukan oleh pengguna pada media berbagi dokumen. Tidak ada operasi yang gagal dicatat pada riwayat Samba.
- ***Full_audit:facility*** merupakan konfigurasi untuk mencatat riwayat Samba dengan menggunakan *syslog*.
- ***Full_audit:priority*** merupakan konfigurasi untuk menentukan tingkat kepentingan riwayat Samba. Adapun tingkatan kepentingan dalam *priority* antara lain *notice*, *info*, *warning*, *debug* dan *alert*.

4.2.2. Konfigurasi *syslogd*

Syslogd merupakan perangkat lunak untuk mencatat riwayat program yang dipasang di sisi server. Konfigurasi yang

ditambahkan pada *syslogd* dapat dilihat pada Gambar 4.2.

```

server@server-VirtualBox: ~/Downloads/Testing2
GNU nano 2.2.6 File: /etc/syslog.conf

# /etc/syslog.conf Configuration file for syslogd.
#
# For more information see syslog.conf(5)
# manpage.
#
##
# First some standard logfiles. Log by facility.
#
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
lpr.* -/var/log/lpr.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
local7.* /var/log/samba/log.audit
#
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 4.2 Konfigurasi syslogd

Konfigurasi yang ditambahkan pada *syslogd* dapat dilihat pada gambar yang diberi warna kuning. Semua riwayat samba dicatat pada data yang berada di */var/log/samba/log.audit*.

4.2.3. Implementasi basis data

Implementasi basis data dengan menggunakan MySQL dijelaskan pada Gambar 4.3. Basis data akan membuat sebuah entitas dengan nama *log_user* yang digunakan untuk penyimpanan riwayat penyalinan dokumen oleh pegawai.

function create table <i>log_user</i>
input : -
output : table <i>log_user</i>
function create table <i>log_user</i> ()
open database
create table <i>log_user</i>
close database
end function

Gambar 4.3 Implementasi Basis Data

4.2.4. Implementasi *filesystem*

Implementasi *filesystem* dengan menggunakan FUSE dijelaskan pada Gambar 4.4. *Filesystem* yang diimplementasikan menggunakan FUSE hanya mampu menjalankan operasi *open* dan *read*. Pada operasi *open* diimplementasikan penambahan informasi ke dalam dokumen Microsoft Office dan dokumen PDF.

```

input: Operation
output: -
class fuse_oper is
    function init is
        input: self is object, root is directory
        output: none
        set self.root = root
        set self.flags = NULL
    end function
    function _full_path is
        input: self is object, path_file is path of file
        output: path
        if path_file.startswith "/"
            path_file = right(:1)
        end if
        path = os.path.join(self.root, path_file)
        return path
    end function
    function getattr is
        input: self is object, path, fh
        output: dict
        full_path = self._full_path(path)
        st = os.lstat(full_path)
        return dict((key, getattr(st, key))
    end function
    function readdir is
        input: self is object, path, fh
        output: -
        full_path = self._full_path(path)

```

```

        dirents = ., ..
        if os.path.isdir full_path

dirents.extends(os.listdir(full_path))
        end if
        for r in dirents
            yield r
        end for
    end function
    function open is
        input: self is object, path, flags
        output: open file
        full_path = self._full_path(path)
        if self.flags != full_path
            if docx in full_path or xlsx in
full_path
                insert = MDO(full_path)
                insert.insert_info()
            end if
            if pdf in full_path
                insert = PDF(full_path)
                insert.insert_info()
            end if
        end if
        return os.open(full_path, flags)
    end function
    function read is
        input: self is object, path, length, offset, fh
        output: read file
        input: self is object path, flags
        os.lseek(fh, offset, os.SEEK_SET)
        return os.read(fh, length)
    end function
end class

```

Gambar 4.4 Pseudocode filesystem

4.2.5. Implementasi penambahan informasi ke dalam dokumen Microsoft Office

Tahapan ini merupakan proses penambahan informasi ke dalam dokumen Microsoft Office. *Pseudocode* proses penambahan informasi ke dalam dokumen Microsoft Office ditunjukkan pada Gambar 4.5.

function insert_info
input: Microsoft Office Document
output: Microsoft Office Document
<pre> function insert_info get timenow get username from samba_log get hash from md5 get xml from get_word_xml get xml_tree from get_xml_tree get xml_format from etree.parse(format) insert into table log_user value username, document_path, timenow, hash index = 0 for element in xml_tree for elements in format elements.text = element.text end for end for index = 0 for element in xml_tree if index = 3 if element.text != NULL element.text = element.text[0:find(-)] set element.text = element.text + '-' + username end if else </pre>

```

set element.text = element.text +
'- ' + username
end if
end if
if index = 5
if element.text != NULL
element.text =
element.text[0:find(-)]
set element.text = element.text +
'- ' + hash
end if
else
set element.text = element.text +
'- ' + hash
end if
end if
if index = 6
if element.text != NULL
element.text =
element.text[0:find(-)]
set element.text = element.text +
'- ' + timenow
end if
else
set element.text = element.text +
'- ' + timenow
end if
end if
index+1
end for
save_docx
end function

```

Gambar 4.5 Pseudocode Penambahan Informasi ke dalam dokumen Microsoft Office

4.2.6. Implementasi penambahan informasi ke dalam dokumen PDF

Tahapan ini merupakan proses penambahan informasi ke dalam dokumen PDF. *Pseudocode* proses penambahan informasi ke dalam dokumen PDF ditunjukkan pada Gambar 4.6.

function insert_info
input: PDF output: PDF
function insert_info get timenow get username from samba_log get hash from md5 insert into table <i>log_user</i> value username, document_path, timenow, hash open_pdf insert timenow, username, hash to metadata PDF save_pdf end function

Gambar 4.6 *Pseudocode* Penambahan Informasi ke dalam dokumen PDF

4.2.7. Implementasi *hash* dokumen menggunakan *md5*

Tahapan ini merupakan proses *hash* dokumen menggunakan algoritma *md5*. *Pseudocode* proses *hash* dokumen menggunakan algoritma *md5* ditunjukkan pada Gambar 4.7.

function md5
input: username, time, document output: hash
function md5 hash1 = md5.hash(document) hash2 = hash1 + username + time hash = md5.hash(hash2) return hash end function

Gambar 4.7 *Pseudocode hash* dokumen menggunakan *md5*

BAB V PENGUJIAN DAN EVALUASI

Pada bab ini akan dibahas mengenai pengujian dari segi fungsionalitas dari aplikasi. Pengujian fungsionalitas akan dibagi ke dalam beberapa skenario pengujian.

5.1. Pengujian Fungsionalitas

Pengujian fungsionalitas merupakan pengujian terhadap jalannya fungsi-fungsi utama aplikasi. Pada aplikasi yang dibangun terdapat tiga fungsional utama. Fungsionalitas tersebut adalah proses menjalankan *filesystem* pada sisi server, menjalankan Samba pada sisi pengguna, penambahan informasi pada Microsoft Dokumen(docx, xlsx dan pptx) dan penambahan informasi pada PDF.

5.1.1. Lingkungan Pengujian

Lingkungan uji coba mencakup perangkat keras dan perangkat lunak. Spesifikasi komputer yang digunakan ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Komputer yang Digunakan

No	Komputer	Keterangan
1	PC 1	Laptop Toshiba dengan prosesor Intel Core i3 2.27 GHz, RAM 4GB, kapasitas Harddisk 500GB.

Sedangkan spesifikasi perangkat lunak yang digunakan adalah sebagai berikut :

- Sistem Operasi berupa Windows 7 Ultimate 64 bit pada *client* dan Sistem Operasi berupa Ubuntu 12.04 64 bit pada *server*,
- Microsoft Visio 2007 untuk melakukan perancangan *flowchart* antarmuka aplikasi,
- Star UML untuk mendesain diagram *usecase*,

- Virtual Box untuk pengembangan sistem dan sebagai sarana untuk membuat server,
- Samba sebagai sarana media berbagi dokumen yang dipasang pada server di dalam Virtual Box,
- pustaka fusepy digunakan untuk membuat *filesystem* berbasis FUSE,
- pustaka lxml digunakan untuk mengolah data XML,
- pustaka pyPdf digunakan untuk mengolah dokumen PDF,
- pustaka ZipFile digunakan untuk mengekstraksi dokumen Microsoft Office,
- MySQL digunakan sebagai basis data untuk menyimpan riwayat penyalinan dokumen oleh pengguna.

5.1.2. Uji Coba Menjalankan *filesystem* pada server

Hasil yang diharapkan dari pengujian adalah *filesystem* dapat berjalan dan dapat diakses oleh pengguna melalui Samba. Skenario pengujian ini dijelaskan pada Tabel 5.2. Gambar 5.1 menunjukkan proses menjalankan *filesystem*. Sedangkan Gambar 5.2 menunjukkan *filesystem* telah berjalan pada server.

Tabel 5.2 Skenario Pengujian Proses Menjalankan *filesystem*

Nama	Menjalankan <i>filesystem</i> pada sisi server
Tujuan	Mengecek keberhasilan menjalankan <i>filesystem</i> pada sisi server.
Kondisi awal	Server belum menjalankan <i>filesystem</i> .
Skenario	Server menjalankan <i>filesystem</i>
Masukan	<i>Filesystem</i> yang diimplementasikan menggunakan fuse dengan bahasa pemrograman python.
Keluaran yang Diharapkan	<i>Filesystem</i> dapat berjalan
Hasil pengujian	Berhasil

```

root@server-VirtualBox: /home/server/Documents
root@server-VirtualBox:/home/server/Documents# python filesystem.py /home/server
/Downloads/ /srv/samba/share/
root@server-VirtualBox:/home/server/Documents#

```

Gambar 5.1 Menjalankan *filesystem*

```

root@server-VirtualBox: /home/server/Documents
root@server-VirtualBox:/home/server/Documents# mount -l
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
gvfs-fuse-daemon on /home/server/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=server)
fuse_oper on /srv/samba/share type fuse (rw,nosuid,nodev,allow_other)
root@server-VirtualBox:/home/server/Documents#

```

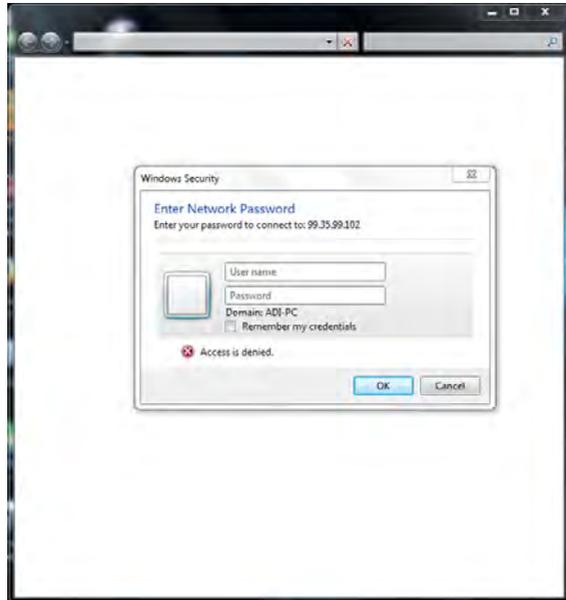
Gambar 5.2 Mengecek *filesystem* pada server

5.1.3. Uji Coba Membuka media berbagi dokumen Samba pada sisi pengguna

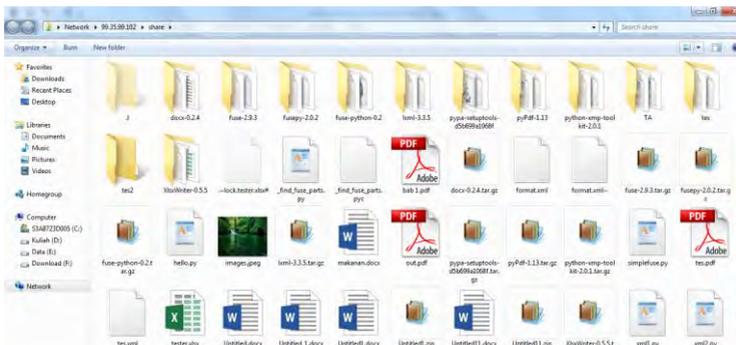
Hasil yang diharapkan dari pengujian adalah media berbagi dokumen Samba dapat diakses oleh pengguna. Skenario pengujian proses membuka media berbagi dokumen Samba dapat dilihat pada Tabel 5.3. Gambar 5.3 menunjukkan proses *login*. Sedangkan Gambar 5.4 menunjukkan media berbagi dokumen dapat diakses oleh pengguna.

Tabel 5.3 Skenario Pengujian Proses Membuka Media Berbagi Dokumen Samba

Nama	Membuka media berbagi dokumen Samba pada sisi pengguna
Tujuan	Mengecek keberhasilan membuka media berbagi dokumen Samba pada sisi pengguna
Kondisi awal	Pengguna membuka media berbagi dokumen.
Skenario	-Pengguna melakukan login. -Pengguna membuka media berbagi dokumen.
Masukan	Username dan Password pengguna
Keluaran yang Diharapkan	Media berbagi dokumen pada sisi server dapat dibuka oleh pengguna
Hasil pengujian	Berhasil



Gambar 5.3 login untuk mengakses Samba



Gambar 5.4 Direktori media berbagi pakai yang dibagikan menggunakan Samba

5.1.4. Uji Coba Penambahan riwayat penyalinan dokumen oleh pengguna pada Basis Data

Pengujian penambahan riwayat penyalinan dokumen oleh pengguna pada Basis Data dilakukan pada sisi server. Seluruh proses penyalinan dokumen yang dilakukan oleh pengguna akan tersimpan pada basis data yang ada di server. Tabel 5.4 menunjukkan skenario uji coba penambahan riwayat penyalinan dokumen oleh pengguna pada Basis Data dan Gambar 5.5 menunjukkan hasil penambahan riwayat penyalinan pada basis data.

Tabel 5.4 Skenario Pengujian Proses Penambahan riwayat penyalinan dokumen oleh pengguna pada basis data

Nama	Penambahan riwayat penyalinan dokumen oleh pengguna pada basis data.
Tujuan	Mengecek keberhasilan penambahan riwayat penyalinan dokumen pada basis data di sisi server.
Kondisi awal	Pengguna menyalin dokumen yang dibagikan menggunakan Samba.
Skenario	Pengguna memilih dokumen yang akan disalin. <i>Filesystem</i> memasukkan informasi ke dalam dokumen yang disalin. Riwayat penyalinan tersebut kemudian disimpan ke dalam basis data.
Masukan	Informasi berupa username, nama dokumen, waktu penyalinan dan hash dokumen.
Keluaran yang Diharapkan	Riwayat penyalinan yang dilakukan oleh pengguna dapat tersimpan pada basis data di sisi server.
Hasil pengujian	Berhasil

```

server@server-VirtualBox: ~
server@server-VirtualBox:~$ sudo mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 609
Server version: 5.5.37-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use log
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from log_user where dokumen = '/pdf1.pdf';
+-----+-----+-----+-----+
| username | dokumen | waktu | hash |
+-----+-----+-----+-----+
| alice | /pdf1.pdf | Jul 11 09:52:14 | 6b14790e96a2ad150a48499548023b2d |
| alice | /pdf1.pdf | Jul 11 09:52:47 | 404d6972a6af0b015abdf762f71b3e23 |
| alice | /pdf1.pdf | Jul 11 09:57:44 | d2210f40ad61917ffb120009209051c1 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

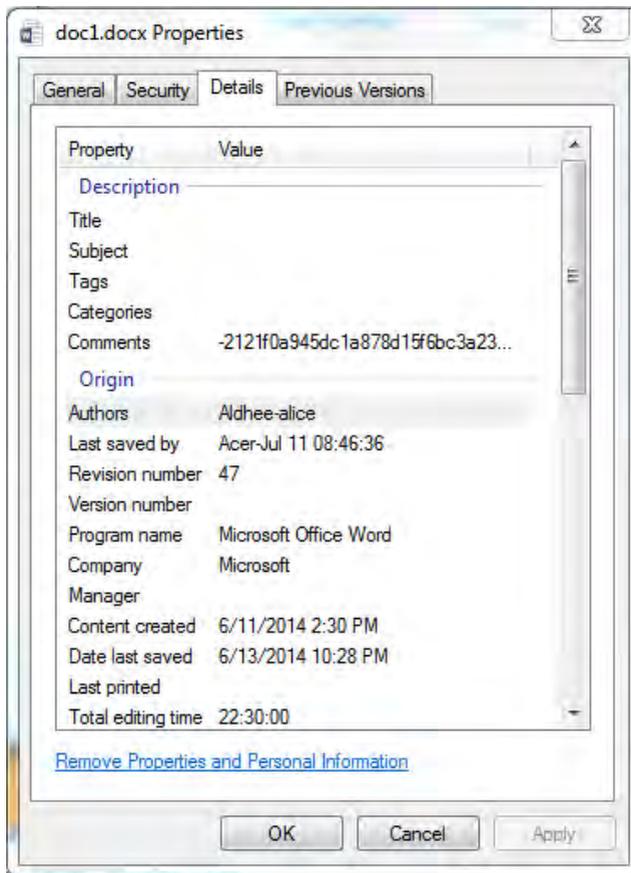
Gambar 5.5 Penambahan riwayat pada Basis Data

5.1.5. Uji Coba Penambahan informasi pada dokumen dengan ekstensi docx(Microsoft Office Word)

Pengujian penambahan informasi pada dokumen dengan ekstensi .docx(Microsoft Office Word) dilakukan pada sisi pengguna. Pengguna akan memilih dokumen yang berekstensi .docx pada media berbagi dokumen Samba yang kemudian dokumen tersebut disalin ke dalam USB *Drive*. Sebelum dokumen tersebut disalin ke dalam USB *Drive*, akan ditambahkan sebuah informasi di dalam properti dokumen tersebut. Informasi tersebut berupa *username* pengguna dan tanggal penyalinan dokumen tersebut oleh pengguna. Tabel 5.5 menunjukkan scenario uji coba proses penambahan informasi pada dokumen dengan ekstensi .docx dan Gambar 5.6 menunjukkan hasil penambahan informasi pada dokumen dengan ekstensi .docx.

Tabel 5.5 Skenario Pengujian Proses Penambahan informasi pada dokumen dengna ekstensi docx

Nama	Penambahan Informasi pada dokumen dengan ekstensi .docx(Microsoft Office Word)
Tujuan	Mengecek keberhasilan penambahan informasi secara transparan pada dokumen dengan ekstensi .docx(Microsoft Office Word).
Kondisi awal	Pengguna membuka media berbagi dokumen.
Skenario	Pengguna memilih dokumen yang berekstensi .docx. Pengguna menyalin dokumen yang berekstensi docx tersebut ke dalam USB <i>Drive</i> .
Masukan	Dokumen berekstensi docx(Microsoft Office Word)
Keluaran yang Diharapkan	Sistem mampu memasukkan informasi pada dokumen yang disalin oleh pengguna ke dalam USB <i>Drive</i> . Dokumen yang disalin bisa dibuka dan dibaca pada USB <i>Drive</i>
Hasil pengujian	Berhasil



Gambar 5.6 Hasil Penambahan informasi pada dokumen dengan ekstensi docx

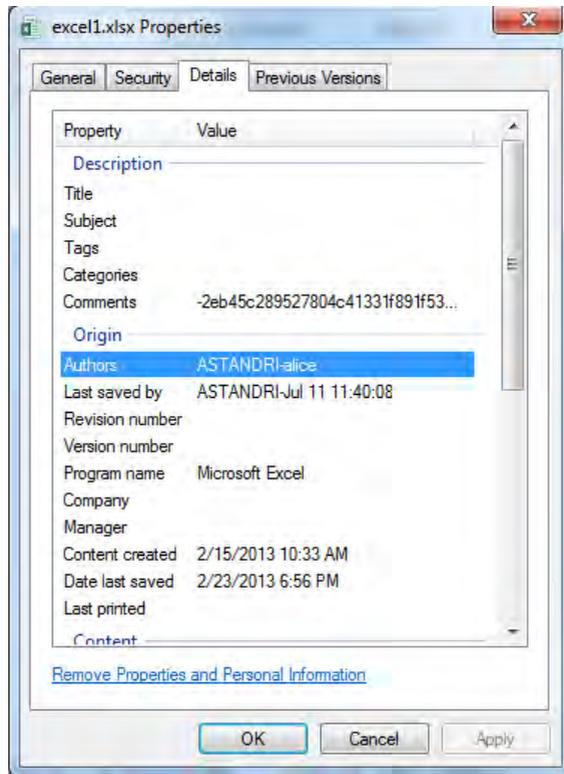
5.1.6. Uji Coba Penambahan informasi pada dokumen dengan ekstensi .xlsx(Microsoft Office Excel)

Pengujian penambahan informasi pada dokumen dengan ekstensi .xlsx(Microsoft Office Excel) dilakukan pada sisi pengguna. Pengguna akan memilih dokumen yang

berekstensi .xlsx pada media berbagi dokumen Samba yang kemudian dokumen tersebut disalin ke dalam USB *Drive*. Sebelum dokumen tersebut disalin ke dalam USB *Drive*, akan ditambahkan sebuah informasi di dalam properti dokumen tersebut. Informasi tersebut berupa *username* pengguna dan tanggal penyalinan dokumen tersebut oleh pengguna. Tabel 5.6 menunjukkan scenario uji coba proses penambahan informasi pada dokumen dengan ekstensi .xlsx dan Gambar 5.7 menunjukkan hasil penambahan informasi pada dokumen dengan ekstensi .xlsx.

Tabel 5.6 Skenario Pengujian Proses Penambahan informasi pada dokumen dengan ekstensi xlsx

Nama	Penambahan Informasi pada dokumen dengan ekstensi xlsx(Microsoft Office Excel)
Tujuan	Mengecek keberhasilan penambahan informasi secara transparan pada dokumen dengan ekstensi xlsx(Microsoft Office Excel).
Kondisi awal	Pengguna membuka media berbagi dokumen.
Skenario	Pengguna memilih dokumen yang berekstensi xlsx. Pengguna menyalin dokumen yang berekstensi xlsx tersebut ke dalam USB <i>Drive</i> .
Masukan	Dokumen berekstensi xlsx(Microsoft Office Word)
Keluaran yang Diharapkan	Sistem mampu memasukkan informasi pada dokumen yang disalin oleh pengguna ke dalam USB <i>Drive</i> . Dokumen yang disalin bisa dibuka dan dibaca pada USB <i>Drive</i>
Hasil pengujian	Berhasil



Gambar 5.7 Hasil Penambahan informasi pada dokumen dengan ekstensi xlsx

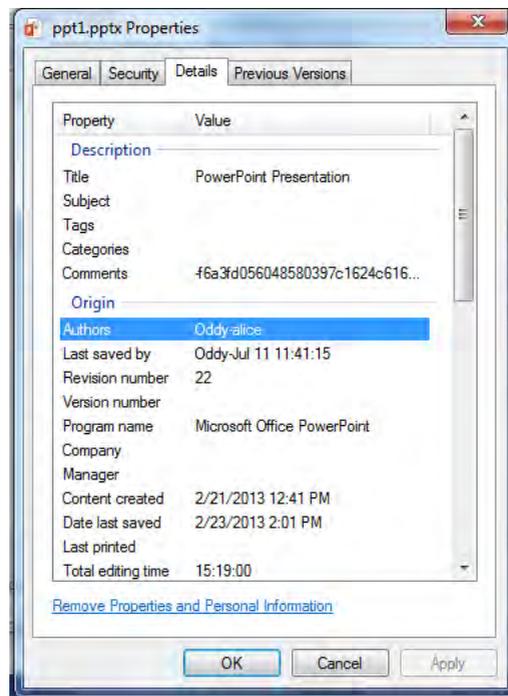
5.1.7. Uji Coba Penambahan informasi pada dokumen dengan ekstensi .pptx(Microsoft Office PowerPoint)

Pengujian penambahan informasi pada dokumen dengan ekstensi .pptx(Microsoft Office PowerPoint) dilakukan pada sisi pengguna. Pengguna akan memilih dokumen yang berekstensi .pptx pada media berbagi dokumen Samba yang kemudian dokumen tersebut disalin ke dalam USB Drive. Sebelum dokumen tersebut disalin ke dalam USB Drive, akan ditambahkan

sebuah informasi di dalam properti dokumen tersebut. Informasi tersebut berupa *username* pengguna dan tanggal penyalinan dokumen tersebut oleh pengguna. Tabel 5.7 menunjukkan scenario uji coba proses penambahan informasi pada dokumen dengan ekstensi .pptx dan Gambar 5.8 menunjukkan hasil penambahan informasi pada dokumen dengan ekstensi .pptx.

Tabel 5.7 Skenario Pengujian Proses Penambahan informasi pada dokumen dengan ekstensi pptx

Nama	Penambahan Informasi pada dokumen dengan ekstensi pptx(Microsoft Office Power Point)
Tujuan	Mengecek keberhasilan penambahan informasi secara transparan pada dokumen dengan ekstensi pptx(Microsoft Office Power Point).
Kondisi awal	Pengguna membuka media berbagi dokumen.
Skenario	Pengguna memilih dokumen yang berekstensi pptx. Pengguna menyalin dokumen yang berekstensi pptx tersebut ke dalam USB <i>Drive</i> .
Masukan	Dokumen berekstensi pptx(Microsoft Office Power Point)
Keluaran yang Diharapkan	Sistem mampu memasukkan informasi pada dokumen yang disalin oleh pengguna ke dalam USB <i>Drive</i> . Dokumen yang disalin bisa dibuka dan dibaca pada USB <i>Drive</i>
Hasil pengujian	Berhasil



Gambar 5.8 Hasil Penambahan informasi pada dokumen dengan ekstensi pptx

5.1.8. Uji Coba Penambahan informasi pada dokumen dengan ekstensi .pdf(Portable Document Format)

Pengujian penambahan informasi pada dokumen dengan ekstensi .pdf(Portable Document Format) dilakukan pada sisi pengguna. Pengguna akan memilih dokumen yang berekstensi .pdf pada media berbagi dokumen Samba yang kemudian dokumen tersebut disalin ke dalam USB *Drive*. Sebelum dokumen tersebut disalin ke dalam USB *Drive*, akan ditambahkan sebuah informasi di dalam properti dokumen tersebut. Informasi tersebut berupa *username* pengguna dan tanggal penyalinan dokumen tersebut oleh pengguna. Tabel 5.8 menunjukkan scenario uji coba proses

penambahan informasi pada dokumen dengan ekstensi .pdf dan Gambar 5.9 menunjukkan hasil penambahan informasi pada dokumen dengan ekstensi .pdf.

Tabel 5.8 Skenario Pengujian Penambahan informasi pada dokumen dengan ekstensi pdf

Nama	Penambahan Informasi pada dokumen dengan ekstensi pdf(Portable Document Format)
Tujuan	Mengecek keberhasilan penambahan informasi secara transparan pada dokumen dengan ekstensi pdf(Portable Document Format).
Kondisi awal	Pengguna membuka media berbagi dokumen.
Skenario	Pengguna memilih dokumen yang berekstensi pdf. Pengguna menyalin dokumen yang berekstensi pdf tersebut ke dalam USB <i>Drive</i> .
Masukan	Dokumen berekstensi pdf(Portable Document Format)
Keluaran yang Diharapkan	Sistem mampu memasukkan informasi pada dokumen yang disalin oleh pengguna ke dalam USB <i>Drive</i> . Dokumen yang disalin bisa dibuka dan dibaca pada USB <i>Drive</i>
Hasil pengujian	Berhasil

```

server@server-VirtualBox: ~/Downloads/Testing2
GNU nano 2.2.6 File: pdf13rd copy.pdf

%PDF-1.3
1 0 obj
<<
/Kids [ 4 0 R 5 0 R 6 0 R 7 0 R 8 0 R 9 0 R ]
/Type /Pages
/Count 6
>>
endobj
2 0 obj
<<
/CreationDate (D\07220140416112756\05307\04700\047)
/Producer (Microsoft\256 Office Word 2007)
/Creator (Microsoft\256 Office Word 2007)
/ModDate (D\07220140416112756\05307\04700\047)
/Comment (alice\055Jul 01 10\07250\07256\0553ee5ab3fe003b53cdfa5c0fe9b4d336d)
/Author (Adi)
>>
endobj
3 0 obj

[ Read 1399 Lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U Uncut Text ^T To Spell

```

Gambar 5.9 Hasil Penambahan informasi pada dokumen dengan ekstensi pdf

5.2. Pengujian Performa

Pengujian performa ini bertujuan untuk mengukur waktu dan perubahan ukuran dokumen pada sistem. Lingkungan uji coba yang digunakan sesuai yang dijelaskan pada bagian 5.1.1. Uji coba untuk mengukur perubahan ukuran dokumen dilakukan pada 20 dokumen digital dan 5 *user dummy*. Daftar 20 dokumen dan 5 *user dummy* dapat dilihat pada Tabel 5.9 dan Tabel 5.10.

Tabel 5.9 Daftar Dokumen yang digunakan untuk Uji Coba

No	Nama Dokumen	Ukuran(dalam byte)
1	doc1.docx	6,719,337
2	doc2.docx	1,830,798
3	doc3.docx	487,483
4	doc4.docx	295,195
5	doc5.docx	20,065
6	excel1.xlsx	19,832
7	excel2.xlsx	15,087
8	excel3.xlsx	12,790
9	excel4.xlsx	13,316

No	Nama Dokumen	Ukuran(dalam byte)
10	excel5.xlsx	17,216
11	ppt1.pptx	449,621
12	ppt2.pptx	597,389
13	ppt3.pptx	59,015
14	ppt4.pptx	49,472
15	ppt5.pptx	77,867
16	pdf1.pdf	217,997
17	pdf2.pdf	697,391
18	pdf3.pdf	4,035,465
19	pdf4.pdf	5,608,148
20	pdf5.pdf	336,696

Tabel 5.10 Daftar *user dummy* yang digunakan untuk Uji Coba

No	<i>Username</i>	<i>Password</i>
1	Alice	Alice
2	Bob	Bob
3	Charlie	Charlie
4	Deltas	Deltas
5	Eminemss	Eminemss

Dan untuk uji coba kecepatan dan waktu transfer dilakukan dengan meyalin dokumen pada 2 kondisi yaitu kondisi pertama tanpa menggunakan *filesystem* dan kondisi kedua menggunakan *filesystem* dengan modul penambahan informasi.

5.2.1. Pengujian Perubahan Ukuran Dokumen

Hasil uji coba pengujian perubahan ukuran dokumen dilakukan penyalinan pada 5 user yaitu pertama dokumen disalin oleh *user* alice, kedua dokumen disalin oleh *user* bob, ketiga dokumen disalin oleh *user* deltas dan yang terakhir dokumen disalin oleh *user* eminemss dan pada 2 kondisi yaitu kondisi pertama menyimpan dokumen Microsoft Office tanpa melakukan kompresi dan dan kondisi kedua dengan melakukan kompresi.

Hasil dari uji coba perubahan dokumen yang disalin oleh user alice pada kondisi pertama ditunjukkan pada Tabel 5.11 dan pada kondisi kedua ditunjukkan pada Tabel 5.16, hasil uji coba perubahan dokumen yang disalin oleh user bob pada kondisi pertama ditunjukkan pada tabel 5.12 dan pada kondisi kedua ditunjukkan pada Tabel 5.17, hasil uji coba perubahan dokumen yang disalin oleh user chalie pada kondisi pertama ditunjukkan pada tabel 5.13 dan pada kondisi kedua ditunjukkan pada Tabel 5.18, hasil uji coba perubahan dokumen yang disalin oleh user deltas pada kondisi pertama ditunjukkan pada Tabel 5.14 dan pada kondisi kedua ditunjukkan pada Tabel 5.19 serta hasil uji coba perubahan dokumen yang disalin oleh user eminenss pada kondisi pertama ditunjukkan pada Tabel 5.15 dan pada kondisi kedua ditunjukkan pada Tabel 5.20. Dokumen doc1.docx dijadikan sampel untuk dibuat grafik perubahan ukuran. Gambar 5.10 menunjukkan grafik perubahan ukuran dokumen doc1.docx pada kondisi pertama dan Gambar 5.11 menunjukkan grafik perubahan ukuran dokumen doc1.docx pada kondisi 2. Perubahan tersebut terjadi karena adanya perbedaan panjang *username*. Tanda (-) pada tabel menunjukkan terjadinya pengurangan ukuran dokumen.

Tabel 5.11 Hasil Perubahan Ukuran Dokumen dengan User alice pada kondisi pertama

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	9,407,110	2,687,773
2	doc2.docx	1,830,798	2,335,877	505,079
3	doc3.docx	487,483	1,222,085	734,602
4	doc4.docx	295,195	849,670	554,475
5	doc5.docx	20,065	123,291	103,226
6	excel1.xlsx	19,832	77,316	57,484
7	excel2.xlsx	15,087	47,950	32,863
8	excel3.xlsx	12,790	36,749	23,959
9	excel4.xlsx	13,316	43,168	29,852

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
10	excel5.xlsx	17,216	68,595	51,379
11	ppt1.pptx	449,621	1,100,024	650,403
12	ppt2.pptx	597,389	721,270	123,881
13	ppt3.pptx	59,015	133,534	74,519
14	ppt4.pptx	49,472	122,132	72,660
15	ppt5.pptx	77,867	203,732	125,865
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.12 Hasil Perubahan Ukuran Dokumen dengan User bob pada kondisi pertama

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	9,407,110	2,687,773
2	doc2.docx	1,830,798	2,335,877	505,079
3	doc3.docx	487,483	1,222,085	734,602
4	doc4.docx	295,195	849,670	554,475
5	doc5.docx	20,065	123,291	103,226
6	excel1.xlsx	19,832	77,316	57,484
7	excel2.xlsx	15,087	47,950	32,863
8	excel3.xlsx	12,790	36,749	23,959
9	excel4.xlsx	13,316	43,168	29,852
10	excel5.xlsx	17,216	68,595	51,379
11	ppt1.pptx	449,621	1,100,024	650,403
12	ppt2.pptx	597,389	721,270	123,881

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
13	ppt3.pptx	59,015	133,534	74,519
14	ppt4.pptx	49,472	122,132	72,660
15	ppt5.pptx	77,867	203,732	125,865
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.13 Hasil Perubahan Ukuran Dokumen dengan User Charlie pada kondisi pertama

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	9,407,110	2,687,773
2	doc2.docx	1,830,798	2,335,877	505,079
3	doc3.docx	487,483	1,222,085	734,602
4	doc4.docx	295,195	849,670	554,475
5	doc5.docx	20,065	123,291	103,226
6	excel1.xlsx	19,832	77,316	57,484
7	excel2.xlsx	15,087	47,950	32,863
8	excel3.xlsx	12,790	36,749	23,959
9	excel4.xlsx	13,316	43,168	29,852
10	excel5.xlsx	17,216	68,595	51,379
11	ppt1.pptx	449,621	1,100,024	650,403
12	ppt2.pptx	597,389	721,270	123,881
13	ppt3.pptx	59,015	133,534	74,519
14	ppt4.pptx	49,472	122,132	72,660
15	ppt5.pptx	77,867	203,732	125,865
16	pdf1.pdf	217,997	217,997	0

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.14 Hasil Perubahan Ukuran Dokumen dengan User deltas pada kondisi pertama

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	9,407,110	2,687,773
2	doc2.docx	1,830,798	2,335,877	505,079
3	doc3.docx	487,483	1,222,085	734,602
4	doc4.docx	295,195	849,670	554,475
5	doc5.docx	20,065	123,291	103,226
6	excel1.xlsx	19,832	77,316	57,484
7	excel2.xlsx	15,087	47,950	32,863
8	excel3.xlsx	12,790	36,749	23,959
9	excel4.xlsx	13,316	43,168	29,852
10	excel5.xlsx	17,216	68,595	51,379
11	ppt1.pptx	449,621	1,100,024	650,403
12	ppt2.pptx	597,389	721,270	123,881
13	ppt3.pptx	59,015	133,534	74,519
14	ppt4.pptx	49,472	122,132	72,660
15	ppt5.pptx	77,867	203,732	125,865
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

**Tabel 5.15 Hasil Perubahan Ukuran Dokumen dengan User
eminemss pada kondisi pertama**

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	9,407,110	2,687,773
2	doc2.docx	1,830,798	2,335,877	505,079
3	doc3.docx	487,483	1,222,085	734,602
4	doc4.docx	295,195	849,670	554,475
5	doc5.docx	20,065	123,291	103,226
6	excel1.xlsx	19,832	77,316	57,484
7	excel2.xlsx	15,087	47,950	32,863
8	excel3.xlsx	12,790	36,749	23,959
9	excel4.xlsx	13,316	43,168	29,852
10	excel5.xlsx	17,216	68,595	51,379
11	ppt1.pptx	449,621	1,100,024	650,403
12	ppt2.pptx	597,389	721,270	123,881
13	ppt3.pptx	59,015	133,534	74,519
14	ppt4.pptx	49,472	122,132	72,660
15	ppt5.pptx	77,867	203,732	125,865
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

**Tabel 5.16 Hasil Perubahan Ukuran Dokumen dengan User alice
pada kondisi kedua**

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	6,634,555	-84,782
2	doc2.docx	1,830,798	762,974	-1,067,824
3	doc3.docx	487,483	474,147	-13,336
4	doc4.docx	295,195	271,973	-23,222
5	doc5.docx	20,065	15,101	-4,964
6	excel1.xlsx	19,832	16,360	-3,472
7	excel2.xlsx	15,087	12,403	-2,684
8	excel3.xlsx	12,790	10,364	-2,426
9	excel4.xlsx	13,316	10,693	-2,623
10	excel5.xlsx	17,216	14,151	-3,065
11	ppt1.pptx	449,621	431,321	-18,300
12	ppt2.pptx	597,389	584,828	-12,561
13	ppt3.pptx	59,015	49,498	-9,517
14	ppt4.pptx	49,472	44,066	-5,406
15	ppt5.pptx	77,867	64,130	-13,737
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

**Tabel 5.17 Hasil Perubahan Ukuran Dokumen dengan User bob
pada kondisi kedua**

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	6,634,553	-84,784
2	doc2.docx	1,830,798	762,972	-1,067,826

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
3	doc3.docx	487,483	474,145	-13,338
4	doc4.docx	295,195	271,971	-23,224
5	doc5.docx	20,065	15,099	-4,966
6	excel1.xlsx	19,832	16,358	-3,474
7	excel2.xlsx	15,087	12,401	-2,686
8	excel3.xlsx	12,790	10,362	-2,428
9	excel4.xlsx	13,316	10,691	-2,625
10	excel5.xlsx	17,216	14,149	-3,067
11	ppt1.pptx	449,621	431,319	-18,302
12	ppt2.pptx	597,389	584,826	-12,563
13	ppt3.pptx	59,015	49,496	-9,519
14	ppt4.pptx	49,472	44,064	-5,408
15	ppt5.pptx	77,867	64,128	-13,739
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.18 Hasil Perubahan Ukuran Dokumen dengan User Charlie pada kondisi kedua

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	6,634,557	-84,780
2	doc2.docx	1,830,798	762,975	-1,067,823
3	doc3.docx	487,483	474,148	-13,335
4	doc4.docx	295,195	271,974	-23,221
5	doc5.docx	20,065	15,102	-4,963
6	excel1.xlsx	19,832	16,361	-3,471

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
7	excel2.xlsx	15,087	12,404	-2,683
8	excel3.xlsx	12,790	10,365	-2,425
9	excel4.xlsx	13,316	10,694	-2,622
10	excel5.xlsx	17,216	14,152	-3,064
11	ppt1.pptx	449,621	431,322	-18,299
12	ppt2.pptx	597,389	584,829	-12,560
13	ppt3.pptx	59,015	49,499	-9,516
14	ppt4.pptx	49,472	44,067	-5,405
15	ppt5.pptx	77,867	64,131	-13,736
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.19 Hasil Perubahan Ukuran Dokumen dengan User deltas pada kondisi pertama

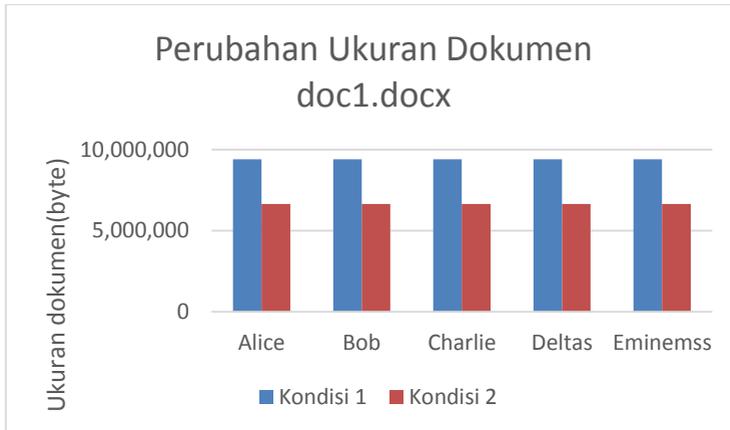
No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	6,634,556	-84,781
2	doc2.docx	1,830,798	762,975	-1,067,823
3	doc3.docx	487,483	474,148	-13,335
4	doc4.docx	295,195	271,974	-23,221
5	doc5.docx	20,065	15,102	-4,963
6	excel1.xlsx	19,832	16,361	-3,471
7	excel2.xlsx	15,087	12,404	-2,683
8	excel3.xlsx	12,790	10,365	-2,425
9	excel4.xlsx	13,316	10,694	-2,622
10	excel5.xlsx	17,216	14,152	-3,064

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
11	ppt1.pptx	449,621	431,322	-18,299
12	ppt2.pptx	597,389	584,829	-12,560
13	ppt3.pptx	59,015	49,499	-9,516
14	ppt4.pptx	49,472	44,067	-5,405
15	ppt5.pptx	77,867	64,131	-13,736
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0

Tabel 5.20 Hasil Perubahan Ukuran Dokumen dengan User eminemss pada kondisi pertama

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
1	doc1.docx	6,719,337	6,634,558	-84,779
2	doc2.docx	1,830,798	762,977	-1,067,821
3	doc3.docx	487,483	474,150	-13,333
4	doc4.docx	295,195	271,976	-23,219
5	doc5.docx	20,065	15,104	-4,961
6	excel1.xlsx	19,832	16,363	-3,469
7	excel2.xlsx	15,087	12,406	-2,681
8	excel3.xlsx	12,790	10,367	-2,423
9	excel4.xlsx	13,316	10,696	-2,620
10	excel5.xlsx	17,216	14,154	-3,062
11	ppt1.pptx	449,621	431,324	-18,297
12	ppt2.pptx	597,389	584,831	-12,558
13	ppt3.pptx	59,015	49,501	-9,514

No	Nama Dokumen	Ukuran Sebelum Penambahan Informasi(byte)	Ukuran Setelah Penambahan Informasi(byte)	Selisih Perubahan
14	ppt4.pptx	49,472	44,069	-5,403
15	ppt5.pptx	77,867	64,133	-13,734
16	pdf1.pdf	217,997	217,997	0
17	pdf2.pdf	697,391	697,391	0
18	pdf3.pdf	4,035,465	4,035,465	0
19	pdf4.pdf	5,608,148	5,608,148	0
20	pdf5.pdf	336,696	336,696	0



Gambar 5.10 Grafik Hasil Perubahan Ukuran Dokumen doc1.docx pada kondisi pertama dan kedua

5.2.2. Pengujian Waktu dan Kecepatan transfer Dokumen

Hasil uji coba pengujian waktu transfer dokumen pada 2 kondisi yaitu tanpa menggunakan *filesystem* dan menggunakan *filesystem* dengan modul penambahan informasi tanpa modul kompresi. Pengujian dilakukan menggunakan dokumen doc1.docx

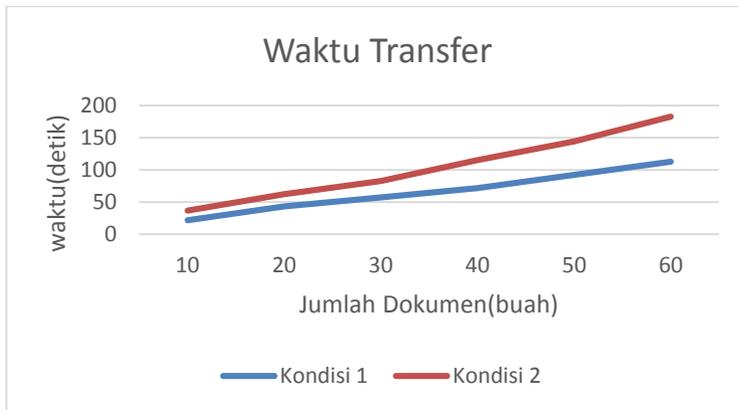
yang berukuran 9,407,110 *byte*. Dokumen tersebut lalu diduplikat sebanyak 60 buah. Kemudian dilakukan pengujian transfer dokumen sebanyak 10 dokumen, 20 dokumen, 30 dokumen, 40 dokumen, 50 dokumen dan 60 dokumen. Hasil dari uji coba dapat dilihat pada tabel 5.21.

Tabel 5.21 Hasil Pengujian waktu transfer dokumen

No	Banyaknya Dokumen	Kondisi 1(dalam detik)	Kondisi 2(dalam detik)
1	10	21.81806	36.81993
2	20	43.05064	62.07386
3	30	57.27149	82.67397
4	40	71.89304	115.187
5	50	92.166	144.4617
6	60	112.4675	182.5813
Jumlah	210	398.6668	623.7978

Dari tabel 5.21 didapatkan rata-rata waktu transfer dokumen pada kondisi 1 yaitu sebesar 1.90 detik dan rata-rata waktu transfer pada kondisi 2 yaitu sebesar 2.97 detik. Sehingga didapatkan perbandingan waktu transfer kondisi 2 1,56 kali lebih cepat dibandingkan waktu transfer kondisi 1. Dapat ditarik kesimpulan waktu yang dibutuhkan untuk melakukan transfer dokumen pada kondisi 1 lebih cepat dibandingkan pada kondisi 2. Grafik waktu transfer dokumen dapat dilihat pada gambar 5.12.

Sedangkan untuk kecepatan transfer didapatkan dari ukuran total dokumen yang disalin dibagi dengan waktu transfer dokumen. Hasil dari kecepatan transfer dokumen dapat dilihat pada tabel 5.22. Penyebab terjadinya perubahan waktu dan kecepatan transfer karena pada *filesystem* yang dibuat terjadi proses membuka dokumen yang akan disalin sehingga menambah waktu proses *filesystem* tersebut.

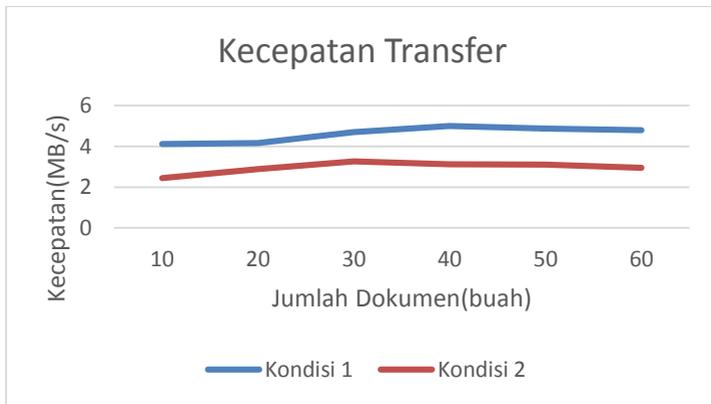


Gambar 5.11 Grafik waktu transfer dokumen

Tabel 5.22 Hasil Pengujian kecepatan transfer dokumen

No	Banyaknya Dokumen	Kondisi 1 (dalam MB/s)	Kondisi 2 (dalam MB/s)
1	10	4.111862	2.43653
2	20	4.167783	2.89052
3	30	4.699348	3.255422
4	40	4.991464	3.115382
5	50	4.866919	3.105075
6	60	4.78607	2.948152
Jumlah	210	27.62345	17.75108

Dari Tabel 5.22 didapatkan rata-rata kecepatan transfer dokumen pada kondisi 1 yaitu sebesar 4.725679 MB/s dan rata-rata waktu transfer pada kondisi 2 yaitu sebesar 3.020163MB/s. Sehingga kecepatan transfer dokumen pada kondisi 1 lebih cepat dibandingkan kecepatan transfer dokumen pada kondisi 2. Grafik waktu transfer dokumen dapat dilihat pada Gambar 5.13.



Gambar 5.12 Grafik kecepatan transfer dokumen

5.2.3. Pengujian Pencarian Riwayat Penyalinan Dokumen

Hasil uji coba pengujian pencarian riwayat penyalinan dokumen dengan skenario 2 user menyalin dokumen yang sama pada waktu yang berbeda. Dokumen yang disalin adalah excel4.xlsx. Gambar 5.13 menunjukkan hasil dari uji coba pengujian pencarian riwayat penyalinan dokumen pada Basis Data. Pada gambar ditunjukkan bahwa dokumen excel4.xlsx disalin oleh orang yang berbeda pada waktu yang berbeda. Dari riwayat yang tercatat tersebut akan dapat diketahui siapa saja orang-orang yang menyalin dokumen tersebut. Sehingga apabila terjadi pada dokumen yang disalin tersebut maka orang-orang tersebutlah yang akan dicurigai membocorkan dokumen tersebut.

```
mysql> select * from log_user where dokumen = '/excel4.xlsx';
+-----+-----+-----+-----+
| username | dokumen      | waktu           | hash                                     |
+-----+-----+-----+-----+
| alice    | /excel4.xlsx | Jul 16 08:11:32 | f3e44e76a0bfdc651abed5aa8c290f56     |
| bob      | /excel4.xlsx | Jul 16 08:16:06 | 94fb709bf6d03586dce759e5464412bb     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Gambar 5.13 Riwayat Penyalinan Dokumen excel4.xlsx pada Basis Data di Server

BAB VI PENUTUP

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir ini serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Kesimpulan yang dapat diambil dalam Tugas Akhir ini adalah sebagai berikut:

1. Untuk mengurangi tingkat kebocoran dokumen dari dalam perusahaan PT.ATI maka dibuat sebuah *filesystem* yang dapat menambahkan informasi ke dalam dokumen yang disalin.
2. *Filesystem* yang dibuat dengan menggunakan FUSE. Di dalam *filesystem* tersebut terdapat modul penambahan informasi ke dalam dokumen Microsoft Office 2007 dengan menggunakan pustaka *zipfile* dan *lxml*. Dan modul penambahan informasi ke dalam dokumen PDF dengan menggunakan pustaka *pyPDF*.
3. Penambahan informasi berupa *user* pegawai dan waktu penyalinan serta *hash* dokumen tersebut menggunakan *md5* pada dokumen Microsoft Office versi 2007 dan PDF berhasil.

6.2. Saran

Dalam pengerjaan Tugas Akhir ini terdapat beberapa saran untuk perbaikan serta pengembangan sistem yang telah dikerjakan sebagai berikut:

1. Penambahan informasi yang dimasukkan ke dalam dokumen.
2. Penambahan jenis dokumen yang bisa dimasukkan informasi menggunakan *filesystem*.
3. Optimasi waktu dan kecepatan transfer dokumen.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Implementasi *filesystem* menggunakan FUSE dengan bahasa pemrograman python

```
import zipfile
import tempfile
import os
import errno
import sys
import shutil
import hashlib
import zlib
import MySQLdb
from lxml import etree
from lxml.etree import Element
from time import strftime, localtime, time
from fuse import FUSE, Operations, FuseOSError
from pyPdf import PdfFileReader, PdfFileWriter
from pyPdf.generic import NameObject,
createStringObject
from libxmp import XMPFiles, consts

class log_user:
    host = 'localhost'
    user = 'root'
    password = 'root'
    db = 'log'

    def __init__(self):
        self.connection =
MySQLdb.connect(self.host, self.user,
self.password, self.db)
        self.cursor = self.connection.cursor()

    def insert(self, query):
        try:
            self.cursor.execute(query)
            self.connection.commit()
        except:
```

```

        self.connection.rollback()

    def query(self, query):
        cursor =
self.connection.cursor( MySQLdb.cursors.DictCu
rsor )
        cursor.execute(query)

        return cursor.fetchall()

    def __del__(self):
        self.connection.close()

class md5:
    def __init__(self, path, user, time):
        self.path = path
        self.user = user
        self.time = time

    def generate_md5(self):
        BLOCKSIZE = 65536
        hasher_doc = hashlib.md5()
        with open(self.path, 'rb') as f:
            buf = f.read(BLOCKSIZE)
            while(len(buf)>0):
                hasher_doc.update(buf)
                buf = f.read(BLOCKSIZE)
        hasher = hashlib.md5()
        hasher.update(hasher_doc.hexdigest())
        hasher.update(self.user)
        hasher.update(self.time)
        return hasher.hexdigest()

class pdf:
    def __init__(self, pdf_filename, path_file):
        self.pdf_filename = pdf_filename
        self.path_file = path_file

```

```

def insert_info(self):
#     GET TIME
            timenow = strftime("%b %d %X",
localtime())
#     -----
#     GET USERNAME
                                username =
file('/var/log/samba/log.audit',
"r").readlines()[-1]
            index1 = username.find('|')
            index2 = username.find('|')
            username = username[index1+3:index2]
#     -----
#     MD5 FUNCTION
                                md5info =
md5(self.pdf_filename,username,timenow)
            file_info = md5info.generate_md5()
#     -----
#     INFO
            info = username + '-' + timenow + '-' +
file_info[0:32]
#     ----
#     INSERT TO DATABASE
            log_users = log_user()
            query = "INSERT INTO log_user VALUES('"
+ username + "','" + self.path_file + "','" +
timenow + "','" + file_info[0:32] + "'"
            log_users.insert(query)
#     -----
#     INSERT INFORMATION TO METADATA IN PDF
DOCUMENT
            fin = open(self.pdf_filename, 'rb')
            pdf = PdfFileReader(fin)
            writer = PdfFileWriter()
            for page in range(pdf.getNumPages()):
                writer.addPage(pdf.getPage(page))
            infoList = writer._info.getObject()
            infos = pdf.documentInfo
            for key in infos:

```

```

        infoList.update({NameObject(key) :
createStringObject(infos[key])})
        infoList.update({NameObject('/Comment') :
createStringObject(info)})
        fout = open(self.pdf_filename+'out.pdf',
'wb')
        writer.write(fout)
        fin.close()
        fout.close()
        os.unlink(self.pdf_filename)
        os.rename(self.pdf_filename+'out.pdf',
self.pdf_filename)

class xml:
    def __init__(self, docx_filename,
path_file):
        self.zipfile =
zipfile.ZipFile(docx_filename)
        self.docx_filename = docx_filename
        self.path_file = path_file

    def get_word_xml(self):
        with open(self.docx_filename) as f:
            xml_content =
self.zipfile.read('docProps/core.xml')
        return xml_content

    def get_xml_tree(self,xml_string):
        return etree.fromstring(xml_string)

    def _write_and_close_docx (self,
xml_content, output_filename):
        try:
            compression = zipfile.ZIP_DEFLATED
        except:
            compression = zipfile.ZIP_STORED
        modes = { zipfile.ZIP_DEFLATED:
'deflated',

```

```

        zipfile.ZIP_STORED:  'stored',
    }
    tmp_dir = tempfile.mkdtemp()
    self.zipfile.extractall(tmp_dir)
    with
open(os.path.join(tmp_dir, 'docProps/core.xml')
, 'w') as f:
        xmlstr = etree.tostring(xml_content,
pretty_print=True)
        f.write(xmlstr)
        filenames = self.zipfile.namelist()
        zip_copy_filename = output_filename
        with zipfile.ZipFile(zip_copy_filename,
"w") as docx:
            for filename in filenames:
                docx.write(os.path.join(tmp_dir,
filename),
filename, compress_type=compression)
            shutil.rmtree(tmp_dir)

    def insert_info(self):
#         GET TIME
                timenow = strftime("%b %d %X",
localtime())
#         -----
#         GET USERNAME
                username =
file('/var/log/samba/log.audit',
"r").readlines()[-1]
                index1 = username.find('|')
                index2 = username.find('|')
                username = username[index1+3:index2]
#         -----
#         MD5 FUNCTION
                md5info =
md5(self.docx_filename, username, timenow)
                file_info = md5info.generate_md5()
                info = file_info[0:32]
#         -----

```

```

#         INSERT TO DATABASE
        log_users = log_user()
        query = "INSERT INTO log_user VALUES('"
+ username + "','" + self.path_file + "','" +
timenow + "','" + file_info[0:32] + "')"
        log_users.insert(query)
#         -----
#         INSERT INFO TO CORE.XML IN OFFICE
DOCUMENT
                                format_filename =
'/home/server/Downloads/format.xml'
        xml_content = self.get_word_xml()
        xml_tree = self.get_xml_tree(xml_content)
        xml_format = etree.parse(format_filename)
        for element in xml_tree.iter():
            tag_tree = element.tag
            element_tree = element.text
            for elements in xml_format.iter():
                tag_format = elements.tag
                if tag_tree == tag_format:
                    elements.text = element_tree
        index = 0
        for element in xml_format.iter():
            if index == 3:
                if '-' in element.text:
                    index1 = element.text.find('-')
                element.text =
element.text[0:index1] + '-' + username
            else:
                element.text = element.text
+ '-' + username
            elif index == 5:
                if '-' in element.text:
                    index1 = element.text.find('-')
                element.text =
element.text[0:index1] + '-' + info
            else:

```

```

        element.text = element.text
+ '-' + info
        elif index == 6:
            if '-' in element.text:
                index1 = element.text.find('-')
            element.text =
element.text[0:index1] + '-' + timenow
            else:
                element.text = element.text
+ '-' + timenow
                index+=1
                self._write_and_close_docx(xml_format,
self.docx_filename)

class fuse_oper(Operations):
    def __init__(self, root):
        self.root = root
        self.flags = ''

    def _full_path(self, path_file):
        if path_file.startswith("/"):
            path_file = path_file[1:]
        path = os.path.join(self.root, path_file)
        return path

    def getattr(self, path, fh=None):
        full_path = self._full_path(path)
        st = os.lstat(full_path)
        return dict((key, getattr(st, key)) for
key in ('st_atime', 'st_ctime',
        'st_gid', 'st_mode',
'st_mtime', 'st_nlink', 'st_size',
'st_uid'))

    def readdir(self, path, fh):
        full_path = self._full_path(path)
        dirents = ['. ', '.. ']
        print 'tes'

```

```
    if os.path.isdir(full_path):
        dirents.extend(os.listdir(full_path))
    for r in dirents:
        yield r

    def open(self, path, flags):
        full_path = self._full_path(path)
        if self.flags != full_path:
            if 'docx' in full_path or 'xlsx' in
full_path or 'pptx' in full_path:
                insert = xml(full_path, path)
                insert.insert_info()
                self.flags = full_path
            if 'pdf' in full_path:
                insert = pdf(full_path, path)
                insert.insert_info()
            return os.open(full_path, flags)

    def read(self, path, length, offset, fh):
        full_path = self._full_path(path)
        os.lseek(fh, offset, os.SEEK_SET)
        return os.read(fh, length)

def main(mountpoint, root):
    FUSE(fuse_oper(root), mountpoint,
allow_other=True)

if __name__ == '__main__':
    main(sys.argv[2], sys.argv[1])
```

BIODATA PENULIS



I Gede Adhiarta Wiandana, biasa dipanggil Adhiarta, dilahirkan di kota Semarang pada tanggal 7 Desember tahun 1992. Penulis adalah anak sulung dari dua bersaudara. Penulis menempuh pendidikan TK Tiga Raja (1997-1998), SDN 2 Sumerta (1998-2004), SMPN 8 Denpasar (2004-2007), SMA Negeri 3 Denpasar (2007-2010). Pada tahun 2010, penulis mengikuti SNMPTN dan diterima di strata satu Jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi

Sepuluh Nopember Surabaya angkatan 2010 yang terdaftar dengan NRP 5110100171. Di Jurusan Teknik Informatika ini, penulis mengambil bidang minat *Net Centric Computing* (NCC) . Selama menempuh kuliah, penulis juga aktif sebagai anggota departemen kewirausahaan di Himpunan Mahasiswa Teknik Computer (HMTC) C-1A. Penulis juga aktif sebagai anggota TPKH-ITS. Pada beberapa acara kampus, penulis juga beberapa kali aktif menjadi panitia, baik sebagai anggota maupun koordinator. Selain itu, penulis beberapa kali menjadi asisten, diantaranya Asisten mata kuliah Organisasi Komputer, Asisten mata kuliah Riset Operasi, dan Asisten Praktikum Sistem Operasi. Penulis dapat dihubungi melalui alamat *e-mail* dragonfly14641@gmail.com.