

# PEMANFAATAN LAYANAN GOOGLE UNTUK MEMBANGUN SERVER FILE ON LINE

Yuli Ambar Kusumawati<sup>1</sup>, R. Rizal Isnanto<sup>2</sup>, Adian Fatchurrochim<sup>2</sup>

## Abstract

Development of information technology today has changed the way of people in completing their work. Professionalism of employees makes them to be able to get jobs done quickly. Efficiency of time and effort required in this regard. If formerly an employee searches data by conventional means, which is to see the data by opening the file documentation, with advances of information technology, it can be done through the Internet. Information systems make all the necessary information neatly organized and supported with an attractive interface making it easier for information seeker.

The design of file servers on-line system is designed to assist an employee in searching of the latest data so they can complete their work accurately and quickly. In searching the data, an employee is just enough to access the Internet. Designing file server system on line is also done in such a way that people who are not interested can not access the data.

By utilizing the resources already available on the internet, in this case Google, it can be a file server on line which is relatively inexpensive, reliable and safe. File servers on-line system can perform the synchronization process, the login process, the process of creating a user account, the process create a file / folder, open the file, delete the file / folder, the file upload process, the download file, rename the file / folder, process to move files / folders, and the process of inserting a file / folder. In the system file server application on line is there is data storage on Google's servers, and the synchronization process serves to make the same metadata that is on google with those in the system application. The permissions on the file server application system on line can be divided into two types, namely as an administrator (root user) has full access rights to the system of this application and as a user who has limited access rights. The root user can perform user settings (user account), to create group and user, and then will set the user enter a part of the group which. Group setting is enabled to share permissions files / folders in the group level.

**Keywords:** Information Systems, On Line Server Files, Google Services

## I. PENDAHULUAN

### 1.1 Latar Belakang

Berkembangnya teknologi informasi mengubah manusia dalam menyelesaikan semua pekerjaannya. Kalau dahulu manusia menyimpan semua informasi dan dokumen menggunakan buku/kertas yang disimpan dalam lemari dokumen maka sekarang berubah menggunakan komputer untuk menyimpan data-data suatu perusahaan.

Sistem aplikasi File On Line ini dirancang sedemikian rupa untuk memudahkan dalam mencari berkas-berkas terkini, dan tentunya berkas-berkas yang tersimpan akan dijaga kerahasiaannya, sehingga yang dapat mengakses berkas tersebut hanya orang-orang yang berkepentingan.

### 1.2 Tujuan

Sistem aplikasi File On Line yang dirancang mempunyai tujuan :

1. Menyajikan file terkini.
2. Memiliki kendali akses atas dokumen, sehingga hanya orang-orang tertentu yang dapat mengakses dokumen tersebut.
3. Meningkatkan produktivitas karyawan dalam menyelesaikan pekerjaannya

### 1.3 Batasan Masalah

Pada Tugas Akhir ini pembahasan akan dibatasi pada hal-hal berikut ini :

1. Sistem aplikasi file on line ini menggunakan fasilitas Google Docs yang bisa diakses melalui Internet.
2. Tugas akhir ini menguraikan implementasi Java sebagai tampilannya.

## II. LANDASAN TEORI

### 2.1 Defenisi Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting.

### 2.2 Alat Bantu Perancangan Sistem

*Unified Modeling Language (UML)* menawarkan sebuah standar untuk merancang suatu model sebuah sistem, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. Dalam perancangan sistem aplikasi file on line ini menggunakan empat diagram UML, yaitu :

1. Diagram Use Case, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dengan sistem.
2. Diagram Aktivitas, menggambarkan berbagai aliran aktivitas dalam sistem dan juga dapat menggambarkan proses paralel yang mungkin terjadi dari beberapa eksekusi.
3. Diagram Komponen, menggambarkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
4. Diagram Urutan, menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu. Biasanya digunakan untuk menggambarkan rangkaian yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

### 2.3 Google Docs

Google Docs merupakan dokumen pengolahan kata dan spreadsheet yang berbasis situs, artinya dokumen, spreadsheet dan presentasi akan tersimpan di situs sehingga dapat diakses di mana pun terkoneksi dengan Internet dan memiliki peramban (*browser*). Google Docs akan menyimpan dokumen terkini dari apa yang dikerjakan dan juga dapat digunakan untuk berbagi dokumen dengan pengguna lain.

### 2.4 Bahasa Pemrograman Java

Penggunaan bahasa pemrograman Java dikarenakan mempunyai beberapa kelebihan, antara lain :

1. Relatif mudah dipelajari
2. Mudah dikembangkan
3. Bebas Platform (*Platform Independent*)

### 2.5 Sekilas tentang Server File On Line

Berkembangnya teknologi informasi mengubah manusia dalam menyelesaikan semua pekerjaannya. Kalau dahulu manusia menyimpan semua informasi dan dokumen menggunakan buku/kertas yang disimpan dalam lemari dokumen maka sekarang berubah menggunakan komputer untuk menyimpan data-data suatu perusahaan.

Perusahaan/instansi membutuhkan satu sistem yang mampu menyediakan informasi yang akurat dan cepat untuk dapat memperlancar pekerjaannya dan tentu saja didukung dengan keamanan dari pengaksesan data tersebut.

Sistem Server File On Line yang dibuat adalah untuk menyajikan file terkini yang dibutuhkan oleh perusahaan/instansi sehingga mempermudah dalam

pencarian data dan tentunya berkas-berkas yang tersimpan akan dijaga kerahasiaannya, sehingga yang dapat mengakses berkas tersebut hanya orang-orang yang berkepentingan.

## III. PERANCANGAN SISTEM

### 3.1 Kebutuhan Sistem

Dalam perancangan sistem aplikasi *file on line* ini diperlukan suatu program yang dapat menghubungkan antara pengguna sistem dengan *Google Documents*. Adapun kebutuhan sistem adalah sebagai berikut :

#### 3.1.1 Account Google

*Account Google* diperlukan untuk dapat masuk ke dalam sistem yang akan memberikan akses ke dalam *Google Documents*. Sistem aplikasi *file on line* ini berjenis aplikasi desktop sehingga otentikasi menggunakan metode *ClientLogin*.

#### 3.1.2 Google Document List Data API

*Google Document List Data API*, selanjutnya disingkat dengan *Google API* saja, merupakan pustaka yang diperlukan oleh pengembang aplikasi klien yang dapat berinteraksi dengan *Google Documents*. *Google API* diperlukan program aplikasi untuk berinteraksi dengan server Google melalui *protocol Google Data API*.

#### 3.1.3 HSQLDB

HSQLDB (*Hyper SQL DataBase*) adalah manajer basisdata relasional yang sesuai dengan standar SQL. HSQLDB termasuk dalam jenis basis data *embedded*. Basis data jenis ini dilekatkan pada program utama dan tidak memerlukan server yang berdiri sendiri yang terpisah dari program utama. Jenis tabel pada HSQLDB ada 3, yaitu : Tabel Memori, Tabel Cache dan Tabel Teks. Sistem aplikasi *file on line* ini menggunakan cara penyimpanan Tabel Cache, dimana hanya sebagian saja dari data atau indeks yang akan disimpan di memori sehingga memungkinkan untuk tabel berkapasitas besar tanpa menghabiskan memori sistem.

#### 3.1.4 Bahasa Pemrograman Java

Pada perancangan sistem aplikasi *file on line* menggunakan bahasa pemrograman Java. Berikut ini hal-hal yang diperlukan untuk menggunakan bahasa pemrograman Java dalam pembuatan aplikasi *file on line* ini;

1. Meng-*instal* Java 1.5 atau lebih tinggi.
2. Mengunduh kebutuhan pustaka untuk klien Google, yaitu *Google Data API*

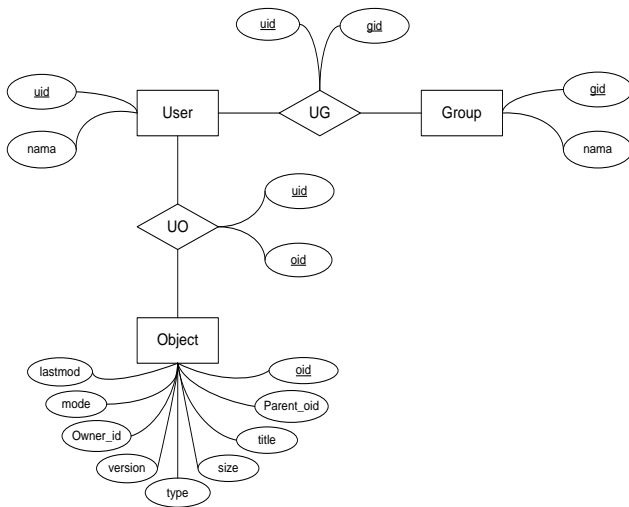
### 3.1.5 NetBeans Platform

NetBeans dikenal sebagai aplikasi IDE (*Integrated Development Environment*) Java untuk mengembangkan aplikasi Java. Inti dari IDE NetBeans Platform adalah sebuah kerangka pengembangan aplikasi berarsitektur *modular* yang memiliki beberapa keuntungan, yaitu :

1. Menyederhanakan penciptaan fitur baru.
2. Memudahkan pengguna untuk menambah dan menghapus fitur.
3. Memudahkan pengguna untuk memperbarui fitur yang ada tanpa mengganggu aplikasi.

### 3.2 Perancangan Basis Data

Tahap perancangan pertama dibuat diagram entitas (Diagram E-R) dari sistem tersebut yang merupakan gambaran relasi antar entitas yang ada dalam sistem. Berikut ini adalah perancangan awal basisdata dari sistem aplikasi *file on line* yang ditunjukkan pada Gambar 3.1.



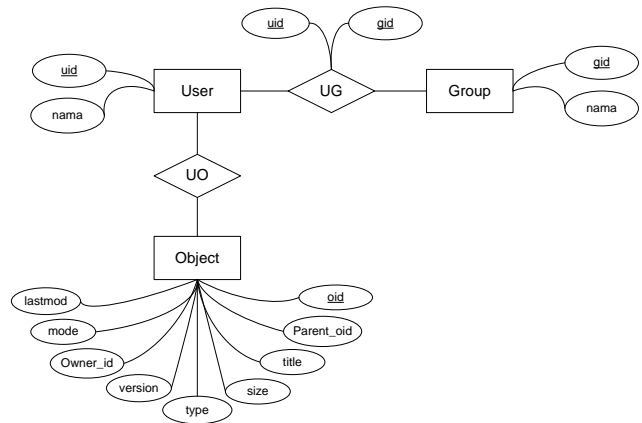
Gambar 3.1 Perancangan Awal Basisdata

Pada Gambar 3.1 dapat dilihat perancangan awal basisdata sistem aplikasi *file on line* yang memiliki tiga entitas, yaitu : **user**, **group** dan **object**. Entitas user mempunyai dua atribut yaitu uid dan nama, sebagai kunci primer adalah **uid**. Entitas Group juga mempunyai dua atribut yaitu gid dan nama, sebagai kunci primer dari group adalah **gid**. Sedangkan untuk entitas object mempunyai sembilan atribut, yaitu : oid, Parent\_oid, title, size, type, version, Owner\_id, mode dan lastmod, sebagai kunci primer adalah **oid**.

Entitas user dan group dihubungkan oleh relasi *usergroup* yang memiliki dua atribut, yaitu : uid dan gid.

Entitas user dan object dihubungkan oleh relasi *uobject* yang memiliki dua atribut, yaitu : uid dan oid. Maksud dari menghubungkan entitas user dengan object yaitu sebuah object dimiliki oleh hanya satu user.

Pada Gambar 3.2 dapat dilihat perancangan akhir basisdata dari sistem aplikasi dengan menghilangkan atribut *uobject* yaitu : uid dan oid. Hal ini dilakukan untuk efisiensi proses query, karena pada rancangan pertama, query untuk mengetahui pemilik object akan melibatkan dua tabel, yaitu : tabel relasi *uobject* dan tabel entitas Object. Selain itu untuk dapat memutakhirkan informasi kepemilikan object juga harus melakukan pemutakhiran di dua tabel, yaitu : tabel *uobject* dan tabel Object.

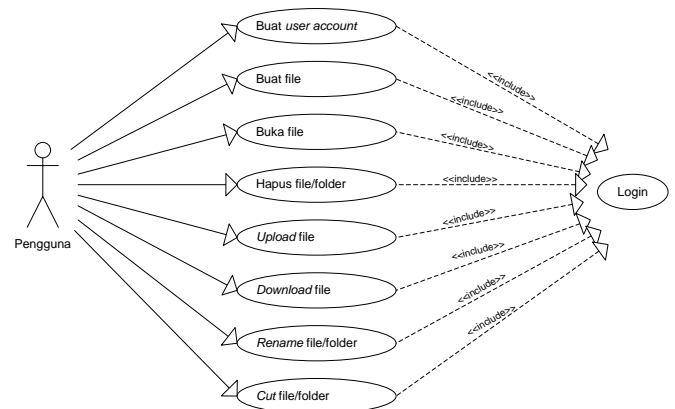


Gambar 3.2 Perancangan Akhir Basisdata

### 3.3 Perancangan Sistem Server File On Line

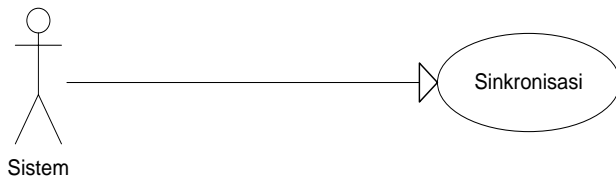
#### 3.3.1 Diagram Use-Case

Pada Gambar 3.3 tampak diagram *use case user* dari aplikasi sistem server *file on line* :



Gambar 3.3 Diagram Use-Case User

Pada Gambar 3.3 menunjukkan diagram *use case user* dengan satu aktor yaitu pengguna (*user*). Berdasarkan gambar tersebut menunjukkan fungsi yang harus dipenuhi oleh sistem agar sistem aplikasi *file on line* dapat berjalan.

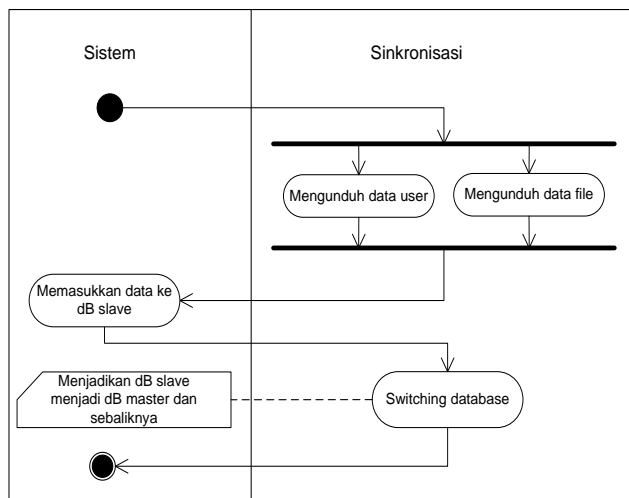


Gambar 3.4 Diagram *Use Case* Sistem

Pada Gambar 3.4 menunjukkan diagram *use case* sistem yang memiliki satu aktor yaitu sistem dan memiliki satu *use case* yaitu sinkronisasi. Berdasarkan gambar tersebut menunjukkan bahwa sistem akan melakukan proses sinkronisasi secara periodik untuk memperbaharui data yang berada di sistem lokal dengan data yang ada di Google.

### 3.3.2 Diagram Aktivitas

Dalam perancangan sistem aplikasi *file on line*, yang menjadi fokus utama adalah sinkronisasi antara sistem aplikasi *file on line* dengan server Google Docs. Gambar 3.5 menunjukkan aktivitas dari sinkronisasi.



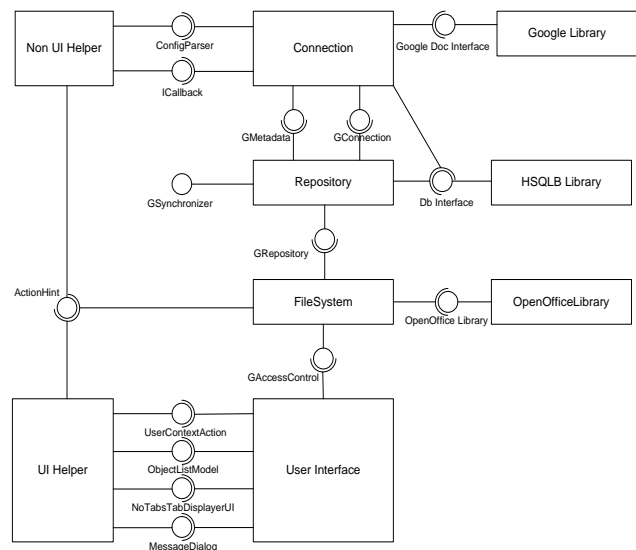
Gambar 3.5 Diagram Aktivitas Sinkronisasi

Pada Gambar 3.5 menunjukkan diagram aktivitas sinkronisasi, dimana sebagai aktornya adalah sistem. Sistem memiliki dua buah basisdata lokal. Basisdata yang pertama disebut dengan basisdata *master* dan yang kedua disebut basisdata *slave*. Pada proses sinkronisasi, basisdata *slave* berfungsi sebagai *buffer* untuk proses yang terjadi. Aktivitas sinkronisasi dimulai oleh sistem dengan mengunduh

data yang berisi informasi pengguna dan daftar objek (*file* dan *folder*) beserta metadatanya. Data ini kemudian dimasukkan ke basisdata *slave*. Setelah proses memasukkan semua data selesai, selanjutnya terjadi proses *switching* basisdata *slave* menjadi basisdata *master* untuk digunakan oleh sub sistem aplikasi lainnya dan basisdata *master* menjadi basisdata *slave* untuk proses sinkronisasi pada periode selanjutnya. Proses sinkronisasi terjadi secara periodik dan paralel dengan sub sistem aplikasi yang berhubungan dengan pengguna, sehingga diharapkan aplikasi secara keseluruhan tetap responsif terhadap perintah yang diberikan pengguna aplikasi.

### 3.3.3 Diagram Komponen

Aplikasi dibangun dengan arsitektur modular yang terdiri dari beberapa modul yang saling berhubungan. Modul adalah sebuah unit tunggal yang *self contained* yang berisi satu atau lebih *class*. Sebuah modul, sedikitnya menyajikan satu antarmuka, yang berupa *class*, yang dapat digunakan oleh modul lainnya untuk berkomunikasi dengan modul tersebut. Gambar 3.6 menunjukkan hubungan antar modul yang ada dalam sistem aplikasi *file on line*.



Gambar 3.6 Diagram Komponen

Pada Gambar 3.6 Diagram Komponen ditunjukkan oleh modul-modul yang ada di sistem aplikasi *file on line*, di antaranya :

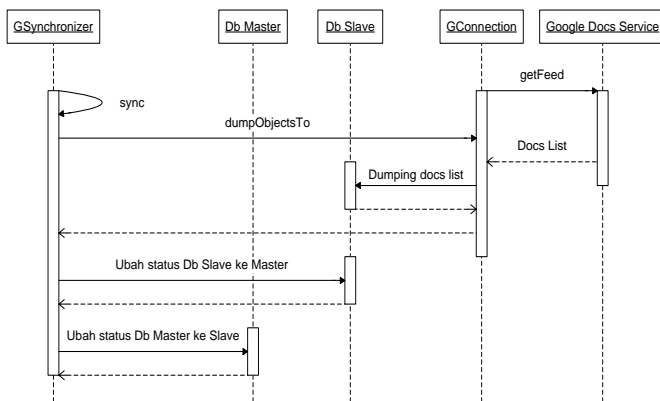
1. Modul *Connection*, modul ini berfungsi sebagai penghubung antara aplikasi dengan server Google.
2. Modul *Repository*, modul ini melakukan sinkronisasi dan menyimpan informasi yang diambil dari server Google Docs ke penyimpanan lokal sebagai *cache* informasi untuk dipakai oleh

bagian aplikasi lainnya. Implementasi penyimpanan lokal berupa dua basisdata, *master* dan *slave*. Basisdata *master* digunakan untuk mendukung operasional aplikasi pada sisi pengguna, sedangkan basisdata *slave* digunakan sebagai penyimpanan sementara pada proses sinkronisasi.

3. Modul *Filesystem*, berfungsi sebagai abstraksi tingkat tinggi dari basisdata *master* menjadi bentuk sistem file yang umum.
4. Modul Antarmuka Pengguna (*User Interface*), berfungsi untuk menampilkan antarmuka pengguna.

### 3.3.4 Diagram Urutan

Diagram urutan digunakan untuk menggambarkan rangkaian yang dilakukan sebagai respons dari sebuah kejadian untuk menghasilkan keluaran tertentu. Gambar 3.7 menunjukkan diagram urutan dari proses sinkronisasi.



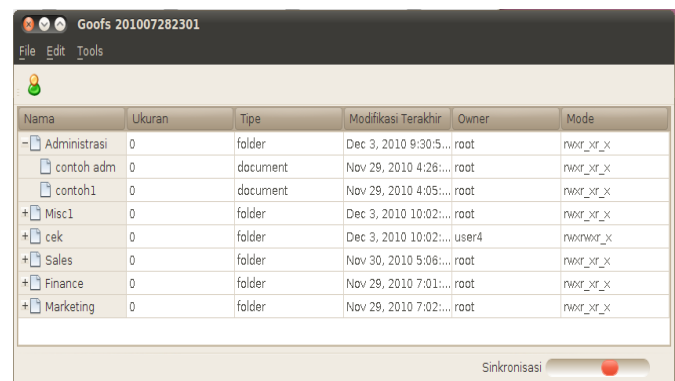
Gambar 3.7 Diagram Urutan Sinkronisasi

Pada proses sinkronisasi, **GSynchronizer** memulai dengan sinkronisasi dengan membuat *thread* baru secara periodik yang didalam *thread* tersebut memanggil metode `private sync` yang ada pada **GSynchronizer**. Kemudian, metode `sync` memanggil metode `dumpObjectsTo` pada **GConnection**, selanjutnya **GConnection** memanggil metode `getFeed` yang ada di **Google Docs Service**. **Google Docs Service** akan memberikan daftar dokumen dan informasi daftar pengguna ke **GConnection** yang kemudian memindahkannya ke **Db Slave**. Setelah proses dumping informasi tersebut selesai, **GSynchronizer** mengubah status basisdata *slave* ke basisdata *master*, dan mengubah status basisdata *master* ke basisdata *slave*.

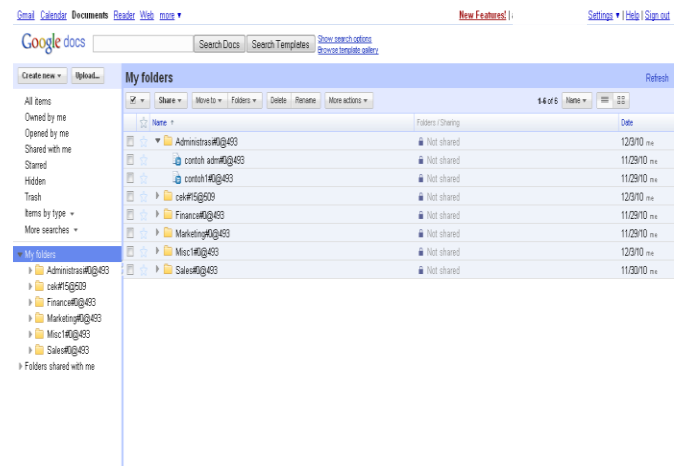
## IV. IMPLEMENTASI DAN PENGUJIAN

Implementasi dan pengujian kali ini meliputi proses koneksi dengan Google Docs dan pengujian proses yang telah dirancang.

Setelah melakukan proses-proses yang terdapat di sistem aplikasi *file on line*, antara lain : login, buat user account, buat file/folder, buka file, hapus file/folder, unggah file/folder, unduh file/folder, ubah nama file/folder, memindah file/folder dan memasukkan file/folder, dapat dilakukan pengecekan bahwa sistem aplikasi *file on line* ini dapat berjalan sesuai dengan perancangan awal. Hal ini dapat dibuktikan dengan cara membuka Google Docs, kemudian login dan membandingkan daftar isi tampilan menu utama dari sistem aplikasi ini dengan daftar isi tampilan menu pada Google Docs. Gambar 4.4 menunjukkan daftar isi tampilan menu utama dari sistem aplikasi *file on line* dan Gambar 4.5 menunjukkan daftar isi tampilan menu pada Google Docs.



Gambar 4.6 Daftar isi tampilan menu utama server file on line



Gambar 4.7 Daftar isi tampilan Google Docs

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Dengan memanfaatkan sumber daya yang telah tersedia di internet, dalam hal ini Google, maka dapat dibuat sebuah sistem aplikasi *file on line* yang relatif murah, dapat diandalkan dan aman.
2. Sistem aplikasi *file on line* ini dapat melakukan proses sinkronisasi, proses login, proses membuat *user account*, proses buat file/folder, proses buka file, proses hapus file/folder, proses unggah file, proses unduh file, proses ubah nama file/folder, proses pindah file/folder, proses memasukkan file/folder.
3. Sistem aplikasi *file on line* ini menyimpan data pada server Google Docs.
4. Proses sinkronisasi pada sistem aplikasi *file on line* ini berfungsi untuk membuat sama metadata yang ada di Google Docs dengan yang ada di sistem aplikasi ini.
5. Hak akses pada sistem aplikasi *file on line* dibedakan menjadi dua jenis, yaitu sebagai administrator (pengguna root) yang memiliki hak akses penuh terhadap sistem aplikasi ini dan sebagai pengguna yang memiliki hak akses terbatas.
6. Pengguna root sebagai administrator dapat melakukan pengaturan pengguna (*user account*), yang berfungsi untuk membuat group dan user, kemudian akan mengatur user masuk menjadi bagian dari group yang mana. Pengaturan group difungsikan untuk membagi hak akses file/folder dalam level group.
7. Pada proses melakukan perubahan file (*edit*), file yang dirubah akan dikunci oleh sistem aplikasi ini agar hanya satu operasi mengubah file (*editing*) yg diijinkan.

### 5.2 Saran

Untuk pengembangan aplikasi selanjutnya, dapat ditambahkan fitur-fitur sebagai berikut;

1. Perlu ditambahkan fasilitas Enkripsi otomatis pada file yang disimpan di server Google agar sistem lebih terjamin keamanan datanya.
2. Perlu ditambahkan fasilitas Dekripsi otomatis saat file yang disimpan di server Google diunduh ke komputer lokal.
3. Perlu ditambahkan fasilitas Kompresi otomatis file/folder baik saat mengunduh atau mengunggah file.
4. Perlu ditambahkan fasilitas mengunggah atau mengunduh file secara massal (*bulk upload/download*) untuk menghemat waktu proses

mengunggah dan mengunduh jika akan memproses banyak file .

5. Perlu ditambahkan fasilitas komputasi akses kontrol yang lebih berdaya guna dengan menambahkan fasilitas kelompok grup (*nested group*).

### DAFTAR PUSTAKA

- [1] Eckel, B., *Thinking in Java, Fourth Edition*, Prentice Hall, United States of America, 2006
- [2] Zukowski, J., *The Definitive Guide to Java Swing, Third Edition*, Apress, Inc, United States of America, 2005
- [3] Horstmann, C., *Core Java Volume I - Fundamentals, Eight Edition*, Prentice Hall, United States of America, 2008
- [4] Boudreau, T., *Rich Client Programming*, Pearson Education, Inc, United States of America, 2007
- [5] Bock, H., *The Definitive Guide to Netbeans Platform*, Apress, Inc, United States of America, 2009
- [6] Blaine, S., *HyperSQL User Guide, Revision 3201*, The HSQL Development Group, 2009
- [7] Kimmel, P., *UML Demystified*, The McGraw-Hill Companies, United States of America, 2005
- [8] Protocol Guide v3.0 - Google Documents List Data API, [http://code.google.com/apis/documents/docs/protocol\\_guide.html](http://code.google.com/apis/documents/docs/protocol_guide.html), 16 September 2010
- [9] Reference Guide v3.0 - Google Documents List Data API, <http://code.google.com/apis/documents/docs/3.0/reference.html>, 16 September 2010
- [10] Java Language Guide v3.0 - Google Documents List Data API, [http://code.google.com/apis/documents/docs/3.0/devlopers\\_guide\\_java.html](http://code.google.com/apis/documents/docs/3.0/devlopers_guide_java.html), 16 September 2010



**Yuli Ambar Kusumawati**  
(L2F305253)

Lahir di Surabaya, 17 Juli 1974

Mahasiswa Teknik Elektro  
Ekstensi 2005,

Bidang Konsentrasi Informatika  
dan Komputer.

Universitas Diponegoro.

Email : yak\_cus@yahoo.com

Semarang, Januari 2011

Menyetujui :

Pembimbing I,

R. Rizal Isnanto, S.T., M.M., M.T.  
NIP. 197007272000121001

Pembimbing II,

Adian Fatchurrochim, S.T., M.T.  
NIP. 197302261998021001