IMPLEMENTASI DAN MONITORING LAYANAN CLOUD STORAGE SEAFILE DENGAN NAGIOS DI FAKULTAS ILMU TERAPAN

IMPLEMENTATION AND MONITORING SERVICE CLOUD STORAGE SEAFILE WITH NAGIOS IN SCHOOL OF APPLIED SCIENCE

Nico Reinaldi[1], Fitri Susanti[2], Tedi Gunawan[3]

[1,2,3]Prodi D3 Teknik Komputer, Fakultas Ilmu Terapan, Universitas Telkom
[1]nicoreinaldi@students.telkomuniversity.ac.id, [2]fitri.susanti@telkomuniveristy.co.id,
[3]tedigunawan@telkomuniversity.ac.id

Abstrak

***Cloud computing* merupakan sebuah layanan *platform, software* ataupun *infrastruktur* yang dapat digunakan oleh banyak orang kapanpun dan dimanapun menggunakan internet. Sekarang ini, teknologi *cloud computing* semakin berkembang terutama dalam layanan *infrastruktur*nya, yaitu menyediakan media penyimpanan yang dapat digunakan oleh *user* untuk menyimpan data tanpa perlu takut data itu hilang.**

Salah satu penyedia layanan media penyimpanan yaitu *Seafile*. *Seafile* merupakan layanan media penyimpanan *open source* dan tidak berbayar. *Seafile* memiliki *feature* perlindungan privasi, diskusi untuk kolaborasi *teamwork* atau instansi, *sinkronisasi file* dan *sharing*, serta *library* yang digunakan untuk menyusun *file*. Sehingga dapat menangani penyimpanan berkas yang lebih teratur, sistem privasi yang aman, dan menghubungkan *file* kepada seluruh *user*.

Layanan *cloud computing* harus dapat diakses dan berjalan dengan baik tanpa mengalami gangguan-gangguan seperti *server down*, maka dari itu diperlukan *Monitoring performance server* sehingga kinerja dari *server* dapat berjalan dengan normal. Fungsi dari *server monitoring* ini untuk mengecek jika terdapat suatu gangguan atau kesalahan yang mengakibatkan penurunan dari performa *server* saat sedang dijalankan. *Monitoring* juga dapat membantu tugas seorang *administrator* dalam memantau *server* secara *remote* dimanapun dia berada dan dapat menerima *notifikasi* jika *server* sedang mengalami masalah.

Kata kunci : *Monitoring, Seafile, Cloud Computing,* Penyimpanan

Abstract

Cloud computing is a service platform, software and infrastructure that can be used by many people anytime, anywhere using the Internet. Today, cloud computing technology is growing, especially in the infrastructure services, namely providing storage media that can be used by users to store data without fear of data was lost. One service provider that is Seafile storage media. Seafile an open source storage media services and not paid. Seafile have a privacy protection feature, discussions to teamwork or agency collaboration, file synchronization and sharing, as well as the library used to compile the file. So it can handle more regular file storage, secure privacy systems, and linking files to all users.

Cloud computing services should be accessible and running properly without experiencing disruptions such as server down, then it is necessary Monitoring server performance so that the performance of servers running normally. This monitoring function of the server to check if there is an interruption or error that resulted in a decrease of performance when the server is running. Monitoring can also help the task of an administrator to monitor the server remotely wherever he is and be able to receive notifications if the server is experiencing problems.

Keywords: Monitoring, Seafile, Cloud Computing, Storage

1. Pendahuluan [10 pts/Bold]

1.1. Latar Belakang

Cloud computing adalah sebuah layanan *platform*, *software* ataupun infrastruktur yang dapat digunakan oleh banyak orang kapanpun dan dimanapun. Sekarang ini teknologi *cloud computing* semakin berkembang terutama

dalam layanan infrastrukturnya yaitu menyediakan *storage* yang dapat digunakan oleh orang untuk menyimpan data, tanpa perlu takut data itu hilang ataupun *diformat* secara tidak sengaja.

Pada saat ini Fakultas Ilmu Terapan membutuhkan penyimpanan suatu berkas agar berkas tidak hilang serta mengatur berkas secara tersusun, serta untuk *sharing* berkas hanya pada satu Fakultas Ilmu Terapan tanpa adanya pihak ke 3 yang dapat mengakses berkas sehingga berkas tersebut aman. Oleh karena itu dibutuhkannya layanan *cloud computing* yang menyediakan *storage* yang bisa menangani perlindungan privasi, serta *sharing* data dan sinkronisasi.

Salah satu penyedia layanan *cloud computing* yang menyediakan *storage* adalah *Seafile*. *Seafile* merupakan layanan *cloud storage opensource* dengan *feature* perlindungan privasi, diskusi untuk kolaborasi *teamwork* atau instansi, sinkronisasi *file* dan *sharing*, serta *library* yang digunakan untuk menyusun *file*. Sehingga dapat menangani penyimpanan berkas yang lebih teratur, sistem privasi yang aman, dan menghubungkan *file* kepada seluruh *user* di Fakultas Ilmu Terapan. Agar dapat mengoptimalkan kinerja *Seafile* maka dibutukannya sebuah *monitoring*.

Salah satu penyedia layanan *monitoring* adalah *Nagios*. *Nagios* adalah sebuah sistem *monitoring opensource* yang digunakan untuk *memonitoring service* dan *host*, serta mengidentifikasi masalah yang ada dalam suatu sistem dan menyelesaikan masalah tersebut. Sehingga kinerja *Seafile* dapat berjalan dengan optimal.

1.2. Rumusan Masalah

Adapun rumusan masalah pada latar belakang yaitu :

1. Bagaimana cara mengatur berkas di server agar terstruktur ?
2. Bagaimana cara mengatur perlindungan privasi pada saat menggunakan *Seafile* ?
3. Bagaimana cara *sharing file* di *server* kepada seluruh *user* di Fakultas Ilmu Terapan ?
4. Bagaimana cara mengoptimalkan kinerja server ?
5. Bagaimana cara mengetahui keadaan server ketika berada jauh dari server ?

1.3. Tujuan

Adapun tujuan dari proyek akhir yang dibuat yaitu :

1. Sistem *library* untuk menyimpan berkas agar terstruktur
2. Keamanan sistem *library* sehingga *file* hanya dapat di akses oleh *user* itu sendiri.
3. Memberi hak akses *file*, sehingga *file* dapat di-*share* keseluruh *user* di Fakultas Ilmu Terapan. Sesuai dengan hak akses masing-masing.
4. *Memonitor host* dan *service* dari layanan *Seafile* sehingga apabila terjadi *down* pada *server* dapat segera diperbaiki.
5. Membuat notifikasi *email* untuk *Nagios* apabila terjadi *down* pada *server* dapat diberitahukan melalui *email*.

1.4. Batasan Masalah

Adapun Batasan masalah yaitu :

1. *Client* akan *login* setelah diberi hak akses oleh *server*.
2. Melakukan *upload*, *download*, *message*, dan *sharing* dari *server* ke *client*.
3. *User Nagios* untuk *memonitoring host* dan *service* dari *server*.
4. *IP server* harus *diset* statis agar tidak berubah.
5. Kapasitas storage untuk setiap *user* yaitu 5GB

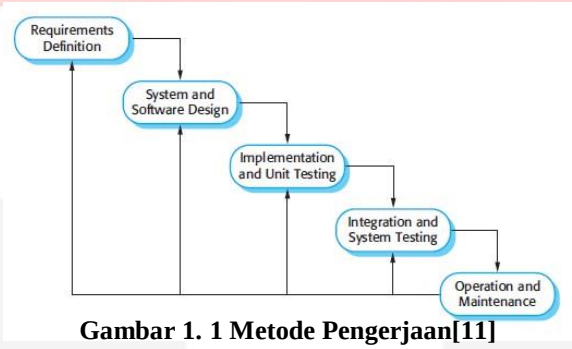
6. Jumlah *user cloud storage Seafile* maksimal hanya 3 sudah termasuk admin

1.5. Definisi Operasional

Mengimplementasikan layanan *cloud storage* pada Fakultas Ilmu Terapan yang dapat diakses oleh semua mahasiswa atau staf dimana saja dan kapan saja. Serta *memonitoring service* dan *host* dari layanan *Seafile* sehingga bisa bekerja lebih optimal.

1.6. Metode Pengerjaan

Adapun metode pengerjaan yang digunakan dalam pengerjaan proyek akhir ini, yaitu *System Development Life Cycle*(SDLC) dengan metode *waterfall*. Metode ini digunakan untuk mengembangkan perangkat lunak yang dibuat secara bertahap, serta apabila terjadi kesalahan bisa kembali ketahap sebelumnya dan tidak perlu mengulangnya dari awal.



Gambar 1. 1 Metode Pengerjaan[11]

1.6.1. Requirements Definition

Pada tahap ini, dianalisa kebutuhan apa saja yang harus disiapkan untuk menjalankan sistem berdasarkan sumber-sumber yang terkait.

1.6.2. System and Software Design

Pada tahap ini, merancang sistem yang akan dibuat berdasarkan analisa dari kebutuhan sistem sebelumnya. Serta membantu dalam menentukan *hardware* dan *system requirements* dan juga membantu mendefinisikan arsitektur dari sistem yang akan dibuat.

1.6.3. Implementation and Unit Testing

Pada tahap ini, pengembangan dari design sistem kedalam program kecil setelah di uji coba kemudian dikembangkan ke tahap berikutnya.

1.6.4. Integration and System Testing

Pada tahap ini, pengembangan dari tahap implementasi kemudian diintegrasikan kedalam sistem setelah diuji coba.

1.6.5. Operations and Maintenance

Pada tahap ini, pengujian sistem yang akan digunakan dalam lingkungan Fakultas Ilmu Terapan.

2. Tinjauan Pustaka

2.1. Cloud Computing

Cloud Computing adalah sebuah model komputasi yang membuat *resource Infomation and Technology (IT)* seperti banyak *server*, *middleware*, dan aplikasi mudah di akses melalui internet kapanun sebagai layanan yang dapat digunakan secara umum atau *private* [1].

Cloud computing di bagi 3 jenis, berdasarkan model penyampaian *(delivery model)*, diantaranya adalah :

1. Software as a Service

   Software As A Service (SAAS) adalah sebuah layanan dimana *user* hanya diberikan layanan aplikasi. Adapun sumber daya yang diberikan sudah dibatasi oleh penyedia layanan sesuai dengan kebutuhan aplikasi yang digunakan oleh *user*. Contohnya adalah google apps, Microsoft dynamics, dll. [2]

2. *Platform as a Service*

   Platform As A Service (PAAS) adalah jenis layanan diatas SAAS dimana *user* diberikan hak untuk mengakses komponen-komponen yang dibutuhkan untuk mengembangkan aplikasinya melalui internet. Jika dianalogikan, PAAS memungkinkan *user* menyewa "rumah" beserta lingkungannya (sistem operasi, *networks*, *database engine*, *framework* aplikasi, dll). [2]

3. *Infrastructur as a Service*

   Infrastructure As A Service (IAAS) adalah sebuah layanan dimana *user* diberikan hak untuk menyewa layanan berupa sumber daya atau infrastruktur secara penuh seperti *proccessor*, *memory*, *storage* dan *banwitdth*. Oleh karena itu *user* dapat memanfaatkan seluruh sumber daya secara penuh sesuai dengan yang *user* beli dari penyedia layanan IAAS seperti Amazon EC2 dan Windows Azure.[2]

   Jika ada kekurangan *resources* (sumber daya), baik itu *procecessor*, *harddisk* maupun *memory*, sehingga dapat dengan mudah sekali menambahkan *server* tambahan dan langsung dapat berintegrasi ke jaringan *cloud*. Butuh waktu waktu sekitar 20 menit untuk menyiapkan *server* kosong atau baru untuk dapat berintegrasi ke jaringan *cloud*.[2] Itulah yang menyebabkan pada saat ini banyak penyedia layanan *hosting* beralih menggunakan teknologi *cloud*.

2.2. Seafile

Seafile adalah sistem *open source cloud storage* dengan *feature* untuk sinkronisasi *file*, perlindungan privasi, dan kolaborasi *teamwork*. Koleksi *file* disebut *library*, dan masing-masing *library* dapat disinkronisasikan secara terpisah. Sebuah *library* bisa dienkripsi dengan *password user* yang dipilih. Sandi tidak disimpan di *server*, sehingga *admin server* tidak dapat melihat isi *file* itu. *Seafile* memungkinkan *user* membuat grup dengan sinkronisasi *file*, dan diskusi untuk memungkinkan kemudahan dalam kolaborasi dalam team.[3]

Server Seafile terdiri dari komponen-komponen berikut :

1. Seahub (Django): Secara default, Seahub berjalan sebagai aplikasi dalam gunicorn. Anda juga dapat mengkonfigurasi Seahub untuk berjalan di bawah mode fast-cgi, tanpa melalui Nginx atau Apache.[3]
2. Server Seafile (seaf-server): layanan data daemon, menangani file upload/download/sinkronisasi. Seafile Server secara default dapat diakses melalui port 8082. Anda dapat mengkonfigurasi Nginx/Apache pada port 8082.[3]
3. Ccnet Server (ccnet-server): RPC layanan daemon untuk memungkinkan komunikasi antara beberapa komponen. Ccnet hanya digunakan untuk komunikasi internal.[3]

2.3. Cloud Storage

Cloud Storage adalah layanan dimana data jarak jauh dipertahankan, dikelola, dan didukung. Layanan ini tersedia bagi *user* melalui jaringan internet. Hal ini memungkinkan *user* menyimpan *file online* sehingga *user* dapat mengaksesnya dari lokasi manapun melalui internet.[4]

Cloud storage adalah media penyimpanan *online* di mana data disimpan pada *server virtual*, umumnya di buat oleh pihak ketiga bukan di *host dedicated server* [5]. Penyedia *cloud storage* bertanggung jawab untuk menjaga data agar tetap *Available* dan *accessible*. *Cloud storage* di akses melalui *web interface API* (*Application Programming Interface)* atau melalui aplikasi yang menggunakan *API* seperti *cloud desktop storage, cloud storage gateway* atau *web* yang berbasis sistem manajemen konten.

Cloud storage didasarkan pada infrastruktur virtual yang persis seperti *cloud computing*, baik itu dalam hal *accessible interfaces, scalability, multi tenancy* dan *resources* [6]. Kapasitas dalam *cloud storage* tergantung kebutuhan dari *user* untuk memakai *cloud storage* tersebut.

2.4. Nagios

Gambar 2. 1 Nagios

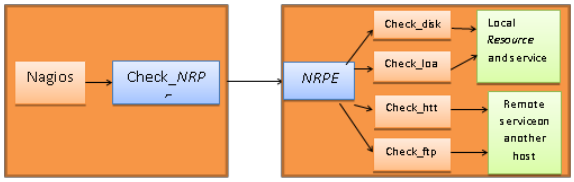
Nagios adalah *tool network monitoring* sistem dengan lisensi *opensource* sehingga bebas untuk digunakan serta dikembangkan. *Nagios* bersifat *modular*, mudah digunakan, dan memiliki skabilitas tinggi. *Modul* atau *plugin* pada *Nagios* sangat sederhana dan dapat membuatnya untuk melengkapi sistem *checking* pada *Nagios* sesuai dengan kebutuhan.[7]

Beberapa fitur yang tersedia pada Nagios diantaranya adalah:

a. *Monitoring* layanan jaringan (*SMTP, HTTP, PING*, dll.)
b. *Monitoring* sumber utama (*Processor load, disk usage*, dll.)
c. Kemampuan untuk mendefinisikan hirarki *host* jaringan dengan menggunakan *"parent"host*, memungkinkan deteksi *host* yang *down* dan *unreachable*.
d. Memberikan pemberitahuan saat *host* mengalami masalah.
e. Pilihan *web interface* untuk menampilkan status jaringan sekarang.

2.5. NRPE (Nagios Remote Plugin Executor)

NRPE dirancang untuk menjalankan *plugin* Nagios pada *remote* mesin Linux/Unix. Alasan utama melakukan ini adalah untuk memungkinkan Nagios untuk memonitor sumber daya lokal (seperti beban *CPU*, *user*an *memory*, dll) pada mesin *remote*. Karena sumber daya umum tidak biasanya di*monitoring* secara langsung, maka agen seperti *NRPE* harus diinstal pada *remote host* Linux/Unix. Berikut adalah ilustrasi dari *NRPE*. [8]



Gambar 2. 2 NRPE (Nagios Remote Plugin Executor)

NRPE terdiri dari dua bagian:

1. *check_NRPE plugin*, yang berada pada mesin *monitoring* lokal.
2. *NRPE daemon*, yang berjalan pada *remote* mesin Linux/Unix.

Ketika Nagios perlu memonitor sumber daya dari *remote* mesin Linux/Unix:

1. Nagios akan mengeksekusi *plugin check_NRPE* dan menanyakan *service* apa saja yang akan di cek.
2. *Plugin check_NRPE* akan menghubungi *NRPE daemon* di *remote host*.
3. *NRPE Daemon* menjalankan *plugin* Nagios yang tepat untuk memeriksa *service*.
4. Hasil dari cek *service* disampaikan dari *NRPE Daemon* kembali ke *plugin check_NRPE*, yang kemudian mengembalikan hasil cek untuk proses Nagios.

2.6. Web Server

Web Server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman web yang ditampilkan dalam web browser. [9]

2.7. Ubuntu

Istilah "*Ubuntu*" adalah sebuah konsep tradisional Afrika yang berasal dari bahasa Bantu Afrika Selatan. Hal ini dapat digambarkan sebagai cara untuk menghubungkan dengan orang lain yang tinggal di sebuah komunitas global di mana tindakan anda mempengaruhi semua umat manusia. *Ubuntu* adalah lebih dari sekedar sebuah sistem operasi: itu adalah komunitas orang-orang yang datang bersama-sama secara sukarela untuk berkolaborasi pada proyek perangkat lunak internasional yang bertujuan untuk memberikan yang terbaik mungkin pengalaman *user*.[10]

Karakteristik *Ubuntu* adalah :

1. *Ubuntu* akan selalu bebas dari biaya, bersama dengan rilis perusahaan *reguler* dan *update* keamanan.[10]
2. *Ubuntu* dilengkapi dengan dukungan komersial penuh dari Canonical dan ratusan perusahaan dari seluruh dunia.[10]
3. *Ubuntu* menyediakan terjemahan terbaik dan aksesibilitas fitur bahwa komunitas perangkat lunak bebas yang ditawarkan.[10]
4. Aplikasi inti Ubuntu semua gratis dan *open source*.[10]

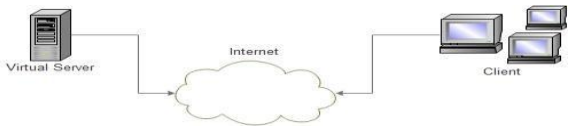
3. Analisis dan Perancangan

3.1. Gambaran Umum Perencanaan Proyek Akhir

Pada proyek akhir ini *cloud storage server* menggunakan *Seafile* dan kemudian *dimonitoring* oleh *nagios* yang akan *diinstal* pada sistem operasi *Ubuntu*. Pada *cloud storage Seafile* akan ada beberapa fungsi seperti pembuatan *library* yang didalamnya terdapat *folder* maupun *file* yang dapat di *share* kepada seluruh *user*, dan juga bisa menambahkan *password* pada *library* yang akan dibuat sehingga *data user* tidak dapat diakses oleh siapapun selain *user* itu sendiri. Pada *Nagios* sudah ada beberapa fungsi yang dibutuhkan untuk melakukan *monitoring server* seperti *monitoring* sumber utama (*Prosesor Load*, *Disk Usage*, *dll*), memberikan pemberitahuan saat *host* mengalami masalah melalui *e-mail*.

Untuk mengimplementasikannya, digunakan satu buah laptop untuk *menginstal* sistem operasi *Ubuntu* yang dimana akan di *instal Seafile* dan *nagios* diatasnya.

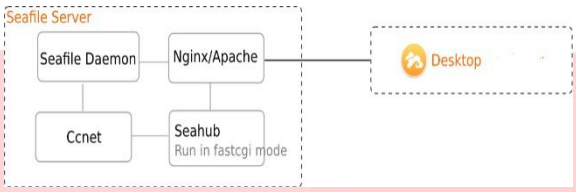
3.1.1. Perancangan Sistem Keseluruhan



Gambar 3. 1 Perancangan Sistem Keseluruhan

Virtual Server merupakan server dengan sistem operasi *Ubuntu 14.04* yang didalamnya sudah *terinstall cloud storage Seafile* dan *Nagios Monitoring*. Client mengakses cloud storage Seafile yang berada pada virtual server dapat melalui internet dengan mengkases ip address server tersebut.

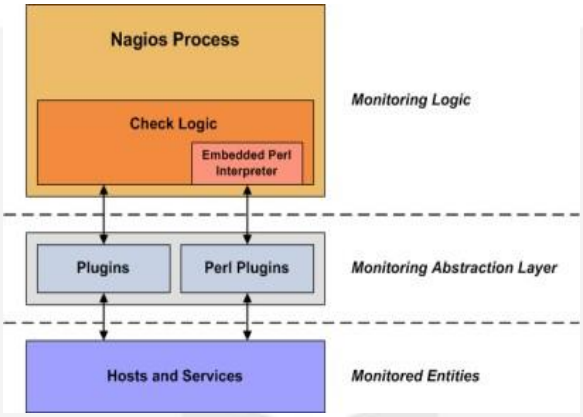
3.1.2. Perancangan Sistem Cloud Storage



Gambar 3. 2 Perancangan Sistem Cloud Storage

Pada perancangan ini *client* dapat mengakses semua layanan *Seafile* termasuk *Seahub* dan *server Seafile* dapat dikonfigurasi melalui ataupun tanpa melalui *Nginx* atau *Apache web server*. Dengan cara ini semua lalu lintas jaringan untuk layanannya bisa *dienkripsi* dengan *HTTPS*. Sehingga *client* bisa terkoneksi dengan baik ke *server*.

3.1.3. Perancangan Sistem Monitoring



Gambar 3. 3 Perancangan Sistem Monitoring

Pada perancangan ini proses *monitoring* pada *nagios* melalui proses pengecekan *plugin* yang sudah disediakan oleh nagios apabila plugin sudah diinstall dapat langsung *memonitoring host* dan service pada *server*.

3.2. Analisis Kebutuhan Sistem

Untuk mendukung agar proyek akhir ini berjalan dengan baik, dibutuhkan spesifikasi perangkat keras dan perangkat lunak yang cukup untuk menjalankan sistem.

3.2.1. Spesifikasi Perangkat Keras dan Perangkat Lunak yang digunakan dalam PC Server

Spesifikasi perangkat keras yang digunakan adalah :

1. *Processor : Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz*
2. *Memory : 2 GB*
3. *Harddisk : 250 GB*

4. *Modem/Wifi*

Spesifikasi perangkat lunak yang digunakan adalah :

1. Sistem Operasi *: Ubuntu 14.04*
2. *LAMPP (Apache, MySQL, PHP)*
3. *Java Runtime Environment 7*
4. *Phyton 2.7.6*
5. *OS Ubuntu 14.04*
6. *Seafile Pro Server 5.1.11*
7. *Nagios Software Monitoring (Nagios Core 4.2.1, Nagios Plugin 2.1.2, Nagios NRPE)*

3.2.2. Spesifikasi Perangkat Keras dan Perangkat Lunak yang digunakan dalam PC Client

Spesifikasi perangkat keras yang digunakan adalah :

1. *Processor : Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz*
2. *Memory : 2 GB*
3. *Modem/Wifi*

Spesifikasi perangkat lunak yang digunakan adalah :

1. Sistem Operasi *: Windows 7*
2. *Web Browser : Mozilla Firefox, Google Chrome, Opera Mini, dll.*
3. *Seafile Desktop5.1.4*

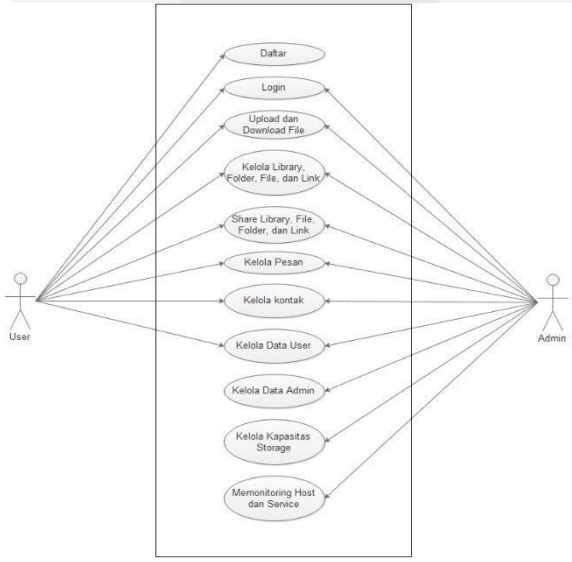
3.3. Database Layanan Cloud Storage Seafile

Cloud storage Seafile yang dibuat menggunakan aplikasi *database MySQL* dan memiliki 3 *database* yang mempunyai fungsinya masing-masing, seperti dibawah ini :

1. ***ccnet-db*** : *database* yang berisi informasi *user* dan *group*
2. ***seafile-db*** : *database* yang berisi perpustakaan *metadata*.
3. ***seahub.db*** : *database* yang berisi tabel yang digunakan oleh *web front end* (*seahub*)

3.4. Perancangan Layanan Pada Server dan Client

Pada Proyek Akhir ini, diperlukan perancangan layanan yang ada pada *server* maupun *client*. Berikut kriteria layanan yang ada pada *admin* maupun *user* :



Gambar 3. 4 Use Case Perancangan Layanan Pada Server dan Client

User dalam *use case* tersebut merupakan *user* yang sudah mendaftar dalam *cloud storage Seafile*.akan tetapi untuk *user admin* ada hak akses khusus yang dapat mengelola kapasitas *storage user*, mengelola data-data *user* yang sudah terdaftar dalam *cloud storage Seafile*, mengelola kontak, pesan, memonitoring layanan, dan lainnya yang sudah disebutkan pada *use case* diatas.

3.4.1. Deskripsi Use Case Perancangan Layanan Pada Server dan Client

1. Deskripsi Layanan Use Case Pada Server

Tabel 3. 1 Deskripsi Layanan Use Case Pada Server

NO.	Use Case	Deskripsi
1.	Login	Proses dimana admin diminta menginputkan email dan password yang sudah terdaftar untuk dapat mengakses halaman admin.
2.	Upload dan Download File	Dalam proses ini admin bisa mendownload ataupun mengupload file.
3.	Kelola Library, Folder, File, dan Link	Dalam proses ini admin bisa membuat dan menghapus library yang didalamnya berisi folder file, file, ataupun sebuah link.
4.	Share Library, File, Folder, dan link.	Dalam proses ini admin bisa melakukan sharing library, file, folder, maupun link ke semua client.
5.	Kelola Pesan	Dalam proses ini admin bisa mengirim pesan, membalas pesan dan menghapus pesan.
6.	Kelola Kontak	Dalam proses ini admin bisa menambah, menghapus daftar kontak.
10.	Kelola Data User	Dalam proses ini admin bisa mengedit, menambahkan, dan menghapus user.
11.	Kelola Data Admin	Dalam proses ini admin bisa mengedit profil.
12.	Kelola Kapasitas Storage	Dalam proses ini admin bisa membatasi kapasitas cloud storage client.
13.	Memonitoring Host dan Service	Dalam proses ini admin memonitoring kinerja dari cloud server. Sehingga apabila ada error pada cloud storage dapat segera diperbaiki.

2. Deskripsi Layanan Use Case Pada Client

Tabel 3. 2 Deskripsi Layanan Use Case Pada Client

NO.	Use Case	Deskripsi
1.	Daftar	Proses registrasi user saat menginputkan Nama, email, kata sandi, dan lain-lain.
2.	Login	Proses dimana user menginputkan email dan password yang sudah didaftarkan.
3.	Upload dan Download File	Dalam proses ini user bisa mendownload ataupun mengupload file.
4.	Kelola Library, Folder, File, dan Link	Dalam proses ini user bisa membuat dan menghapus library yang didalamnya berisi folder file, file, ataupun sebuah link.
5.	Share Library, File, Folder, dan link.	Dalam proses ini user bisa melakukan sharing library, file, folder, maupun link ke semua client.
6.	Kelola Pesan	Dalam proses ini user bisa mengirim pesan, membalas pesan dan menghapus pesan.
7.	Kelola Kontak	Dalam proses ini user bisa menambah, menghapus daftar kontak.
8.	Kelola Data User	Dalam proses ini user bisa mengedit, menambahkan, dan menghapus user.

4. Implementasi dan Pengujian

Pada tahap ini komputer *server* menggunakan *ip 192.168.1.6*. serta ada beberapa tahapan yang akan dijelaskan, yaitu :

1. Instalasi sistem operasi *Ubuntu Desktop 14.04*
2. Instalasi *Cloud Storage Seafile*
3. Instalasi *Nagios Core*
4. Instalasi *Nagios Plugin*
5. Instalasi *Nagios NRPE*
6. Instalasi *mail server Postfix*
7. Konfigurasi *cloud storage Seafile*
8. Konfigurasi *Nagios*
9. Konfigurasi notifikasi *Nagios* menggunakan *mail server*
10. Pengujian

4.1. Instalasi Sistem Operasi Ubuntu Dekstop 14.04

1. Pada tahap ini, *install* sistem operasi *Ubuntu Desktop* seperti biasanya. Akan tetapi memasuki langkah kedua, centanglah "*install this third-party software*". Hal ini sangat penting, karena apabila pada tahap tersebut tidak dicentang, maka *Ubuntu Desktop 14.04* tidak akan bisa menggunakan *Wifi*, dan harus di *update* terlebih dahulu menggunakan koneksi *LAN*.
2. Kemudian *update* dan *upgrade Ubuntu Desktop 14.04*
3. Lalu setelah *update* dan *upgrade install LAMPP (Apache, MySQL, PHP)*

4.2. Instalasi Cloud Storage Seafile

Pada tahap instalasi *cloud storage Seafile* ini menggunakan *Seafile Server Pro 5.1.11* karena memiliki lebih banyak fitur dibandingkan dengan *Seafile Server* lainnya. Berikut cara instalasi *Seafile Server Pro 5.1.11* :

1. Instalasi paket-paket yang diperlukan dalam instalasi *Seafile Server Pro 5.1.11*, menggunakan perintah dibawah ini :

```
root@seafile-nagios:/home/nico/cloudstorage# apt-get install openjdk-7-jre poppler-utils libpython2.7 python-pip \
> mysql-server python-setuptools python-imaging python-mysqldb python-memcache python-ldap \
> python-urllib3
```

Gambar 4. 1 Instalasi Paket untuk Install Seafile

2. *Download* dan extrak *Seafile Server Pro 5.1.11*
3. Masuk ke folder *Seafile Server Pro 5.1.11* yang sudah di extrak
4. Jalankan perintah dibawah ini untuk menginstall *Seafile* melalui *MySQL*

```
root@seafile-nagios:/home/nico/cloudstorage/seafile-pro-server-5.1.11# ./setup-seafile-mysql.sh
```

Gambar 4. 2 Instalasi Seafile Menggunakan MySQL

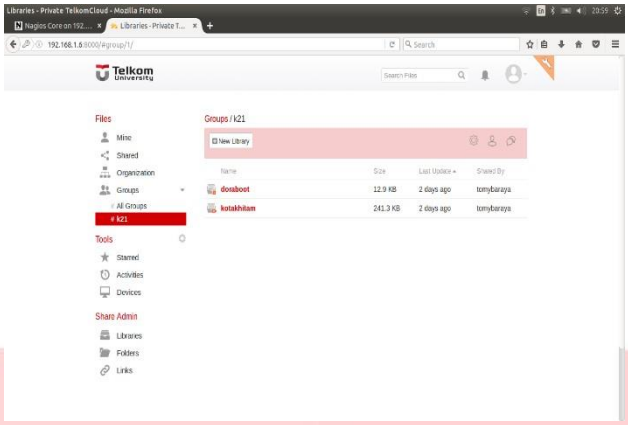
```
What is the name of the server? It will be displayed on the client.
3 - 15 letters or digits
[ server name ] TelkomCloud

What is the ip or domain of the server?
For example: www.mycompany.com, 192.168.1.101
[ This server's ip or domain ] 192.168.1.6

Where do you want to put your seafile data?
Please use a volume with enough free space
[ default "/home/nico/cloudstorage/seafile-data" ]

Which port do you want to use for the seafile fileserver?
[ default "8082" ]

-----------------------------------------------------
Please choose a way to initialize seafile databases:
-----------------------------------------------------

[1] Create new ccnet/seafile/seahub databases
[2] Use existing ccnet/seafile/seahub databases

[ 1 or 2 ] 1

What is the host of mysql server?
[ default "localhost" ]

What is the port of mysql server?
[ default "3306" ]

What is the password of the mysql root user?
[ root password ]

verifying password of user root ...  done

Enter the name for mysql user of seafile. It would be created if not exists.
[ default "root" ] root

Enter the database name for ccnet-server:
[ default "ccnet-db" ]
```

Gambar 4. 3 Input Form Data Seafile

```
---------------------------------
This is your configuration
---------------------------------

    server name:            TelkomCloud
    server ip/domain:       192.168.1.6

    seafile data dir:       /home/nico/cloudstorage/seafile-data
    fileserver port:        8082

    database:               create new
    ccnet database:         ccnet-db
    seafile database:       seafile-db
    seahub database:        seahub-db
    database user:          root



---------------------------------
Press ENTER to continue, or Ctrl-C to abort
---------------------------------
```

Gambar 4. 4 Hasil Setelah Instalasi Seafile

5. *Start service ./Seafile.sh dan ./seahub.sh <port>* untuk menjalankan *Seafile*. *Port service ./seahub.sh default* menggunakan 8000 apabila pada tahap instalasi *port ./seahub.sh* dirubah maka *port ./seahub.sh* pun akan berubah.
6. Masuk ke *web browser* dan jalankan aplikasi *Seafile* dengan mengetikan *http://localhost:8000* atau *http://192.168.1.6:8000*



Gambar 4. 5 Tampilan Web Interface Seafile

4.3. Instalasi Nagios Core

1. Install paket yang diperlukan dalam instalasi *Nagios Monitoring*, seperti perintah dibawah ini :

```
root@seafile-nagios:/home/nico/Monitoring# apt-get install build-essential libgd2-xpm-dev apache2-utils unzip
```

Gambar 4. 6 Instalasi Paket yang Diperlukan untuk Install Nagios

2. Membuat user akun *Nagios* baru di *Ubuntu Desktop 14.04* dengan perintah dibawah ini :

```
root@seafile-nagios:/home/nico/Monitoring# useradd -m nagios
root@seafile-nagios:/home/nico/Monitoring# passwd nagios
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@seafile-nagios:/home/nico/Monitoring# groupadd nagcmd
root@seafile-nagios:/home/nico/Monitoring# usermod -a -G nagcmd nagios
root@seafile-nagios:/home/nico/Monitoring# usermod -a -G nagcmd www-data
root@seafile-nagios:/home/nico/Monitoring#
```

Gambar 4. 7 Membuat User Akun Ubuntu Desktop

3. Download dan extrak *Nagios Core 4.2.1*
4. Masuk ke folder *Nagios Core 4.2.1* yang sudah di extrak
5. *Install* dan *compile Nagios Core 4.2.1* dengan perintah sebagai berikut :

./configure --with-command-group=nagcmd
make all
make install make
install-init make
install-config
make install-commandmode

6. *Install* dan *compile Nagios web interface* ke *Apache*

```
root@seafile-nagios:/home/nico/Monitoring/nagios-4.2.1# /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf
root@seafile-nagios:/home/nico/Monitoring/nagios-4.2.1# sudo ls -l /etc/apache2/sites-enabled/
total 4
lrwxrwxrwx 1 root root   35 Okt  7 11:40 000-default.conf -> ../sites-available/000-default.conf
-rw-r--r-- 1 root root 1679 Okt  7 13:22 nagios.conf
root@seafile-nagios:/home/nico/Monitoring/nagios-4.2.1#
```

Gambar 4. 8 Install dan Compile Nagios Web Interface ke apache

7. Buat akun "*nagiosadmin*" untuk *login* di *Nagios web interface*

```
root@seafile-nagios:/home/nico/Monitoring/nagios-4.2.1# sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
root@seafile-nagios:/home/nico/Monitoring/nagios-4.2.1#
```

Gambar 4. 9 Membuat Akun Nagios Admin Untuk Login di Nagios Web Interface

8. Kemudian *restart apache2 service*

```
service apache2 restart
```

4.4. Instalasi Nagios Plugin

1. *Download* dan extrak *Nagios Plugin 2.1.2*
2. Masuk ke folder *Nagios Plugin 2.1.2* yang sudah di extrak
3. *Install* dan *compile Nagios Plugin 2.1.2* dengan perintah sebagai berikut :

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

4. *Edit file /etc/apache2/site-enabled/Nagios.conf* agar dapat mengakses *ip address 192.168.1.0/24.*

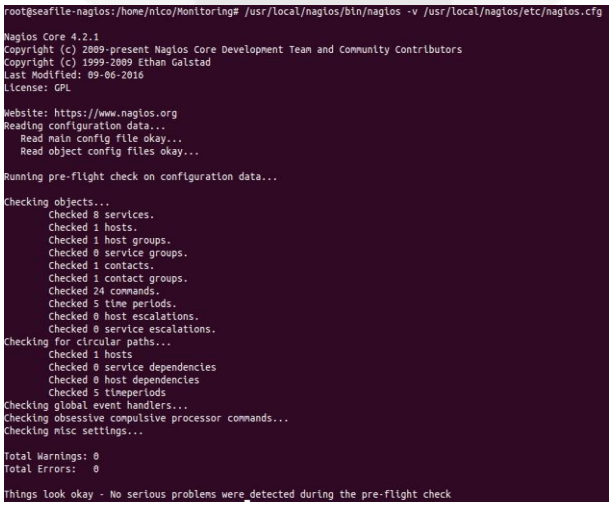
```
[...]
## Comment the following lines ##
#   Order allow,deny
#   Allow from all
## Uncomment and Change lines as shown below ##
Order deny,allow
Deny from all
Allow from 127.0.0.1 192.168.1.0/24
[...]
```

5. *Enable apache rewrite* dan *cgi modules* dengan perintah berikut :

Gambar 4. 10 Enable apache rewrite dan cgi modules

6. *Restart apache2 service*

7. cek apakah ada *error* dalam *syntax Nagios* menggunakan perintah berikut :



Gambar 4. 11 Cek Error Nagios Syntax

8. *Start service Nagios* dengan perintah :

```
root@seafile-nagios:/home/nico/Monitoring# service nagios start
```

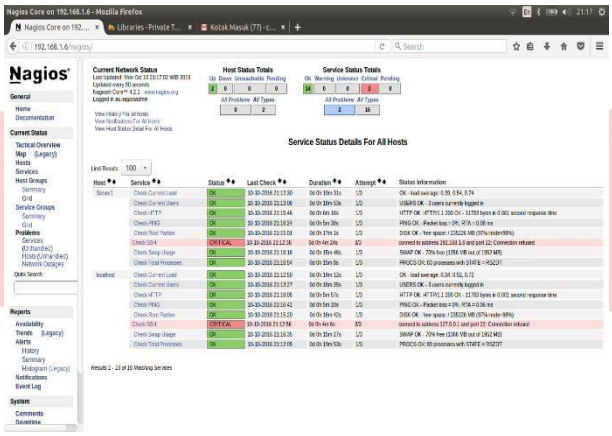
Gambar 4. 12 Start Service Nagios

9. Masukkan perintah agar *Nagios* berjalan otomatis setelah *restart pc* tanpa perlu *start service Nagios* dengan perintah

```
root@seafile-nagios:/home/nico/Monitoring# ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

Gambar 4. 13 Menjalankan Nagios otomatis setelah restart pc

10. Akses *Nagios web interface* melalui *web browser* dengan mengetik *http://localhost/nagios* atau *http://192.168.1.6/nagios*



Gambar 4. 14 Web Interface Nagios

4.5. Instalasi Nagios NRPE (Nagios Remote Plugin Executor)

1. *Install Nagios NRPE Server* dan *Nagios Plugins* dengan perintah berikut

```
root@seafile-nagios:/home/nico/Monitoring# apt-get install nagios-nrpe-server nagios-plugins
```

Gambar 4. 15 Instalasi Nagios NRPE

2. *Edit /etc/nagios/nrpe.cfg* dan tambahkan *ip address* dengan cara berikut ini

Find the following line and add the Nagios server IP
allowed_hosts=127.0.0.1 192.168.1.6

3. *Restart Nagios NRPE Server* dengan perintah berikut

/etc/init.d/nagios-nrpe-server restart

4.6. Instalasi Mail Server Menggunakan Postfix

1. *Install Postfix* dan *mailx* dengan perintah berikut :

```
root@seafile-nagios:/home/nico/Monitoring# apt-get install postfix
```

Gambar 4. 16 Install Postfix

```
root@seafile-nagios:/etc/postfix# apt-get install mailx
```

Gambar 4. 17 Install Mailx

2. *Edit file main.cf* di folder */etc/postfix/* agar menggunakan *smtp gmail*

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = Seafile-nagios
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = localhost

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.1.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = loopback-only
inet_protocols = all
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/gmail_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_use_tls = yes
```

4.7. Konfigurasi Cloud Storage Seafile

Ada beberapa konfigurasi yang dilakukan pada *cloud storage Seafile*, yaitu :

1. Merubah pengaturan default kuota cloud storage dari 2GB menjadi 5GB dengan cara menambahkan perintah dibawah ini ke file *seafile.conf*

```
[quota]
# default user quota in GB, integer only
default = 5
```

2. Merubah pengaturan ukuran maximum upload dan download, dengan menambahkan perintah dibawah ini ke file *seafile.conf*

```
[fileserver]
# Set maximum upload file size to 200M.
max_upload_size=2048
# Set maximum download directory size to 200M.
max_download_dir_size=2048
```

3. Menambahkan fitur pemberitahuan email kepada user yang mendaftar menggunakan *smtp gmail*, dengan menambahkan perintah dibawah ini ke file *seahub_settings.py*

```
EMAIL_USE_TLS = True EMAIL_HOST =
'smtp.gmail.com' EMAIL_HOST_USER =
'cobasaja600@gmail.com'
EMAIL_HOST_PASSWORD = 'password'
EMAIL_PORT = 587
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
SERVER_EMAIL = EMAIL_HOST_USER
```

4. Menambahkan beberapa fitur pada *seafie* seperti *signup*, mengirim *email* verifikasi, mengirim *email* verifikasi *reset password*, mengatur panjang dan kekuatan dari *password* yang digunakan, serta pengaturan lainnya. Dengan cara menambahkan perintah yang ada pada lampiran 1 ke file *seahub_settings.py*
5. Merubah pengaturan logo dan css yang ada pada cloud storage Seafile. Dengan cara menambahkan file css dan logo yang sudah dibuat dan menerapkannya ke *cloud storage Seafile* dengan menambahkan perintah seperti dibawah ini pada *seahub_settings.py*

```
LOGO_PATH = 'custom/logo.png'
BRANDING_CSS = 'custom/custom.css'
```

4.8. Konfigurasi Nagios

Konfigurasi untuk Nagios berupa pembuatan file monitor.cfg untuk menyatukan beberapa file konfigurasi menjadi satu kesatuan, ditulis pada lampiran 2.

4.9. Konfigurasi Notifikasi Nagios Menggunakan Mail Server Postfix

1. Mendefinisikan kontak yang akan menerima notifikasi email, apabila terjadi kesalahan pada host dan service. Dengan menggunakan perintah dibawah ini dan menambahkannnya pada file monitor.cfg

```
define contact {
contact_name            nagiosadmin alias
Nagios Admin service_notification_period non-stop
host_notification_period non-stop
service_notification_options w,u,c,r
host_notification_options d,r
service_notification_commands notify-service-by-email
host_notification_commands notify-host-by-email
email cobasaja600@gmail.com
}
```

2. Memonitoring host ataupun service dengan menambahkan notification interval, dan notification option. Notification interval digunakan untuk mengatur kapan saja Nagios akan mengirimkan email pemberitahuannya ke email yang sudah didefinisi kontak, sedangkan notification option digunakan agar Nagios mengirimkan email saat host maupun service dalam keadaan tertentu. Misalnya saat host dan service mengalami critical, recovery, unknown, dan yang lainnya. Disini contoh perintah yang digunakan untuk memonitoring host agar mendapat notifikasi keadaan host tersebut adalah :

```
define host {
use                          linux-server ; Name of host template to use
host_name                    localhost
alias                        Server PC Nagios
address                 127.0.0.1
check_command                     check-host-alive
max_check_attempts           20
check_period                 non-stop
notification_interval        120
notification_period          non-stop
notification_options         d,u,r
stalking_options             o,d,u
contact_groups               linux-admins
}
```

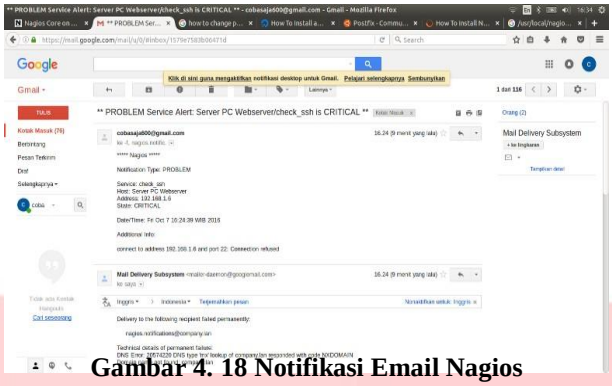
Untuk perintah yang lainnya tulis dalam lampiran 2 di file monitor.cfg

4.10. Pengujian

4.10.1. Pengujian Fungsionalitas Notifikasi Email

Pengujian dilakukan dengan merubah *port ssh server* yang *dimonitoring*, perubahan bisa dilakukan dengan merubah *port ssh* yang *dimonitoring* menggunakan *openssh* . sehingga secara otomatis *notifikasi email* bahwa *service localhost* pada *localhost* dan *server1* mengalami *critical* akan dikirim ke alamat *email* yang sudah didefinisikan saat konfigurasi.

Dibawah ini adalah gambar *notifikasi email* yang menyatakan bahwa *service ssh* mengalami *critical* yang dikirim setiap 2 jam :



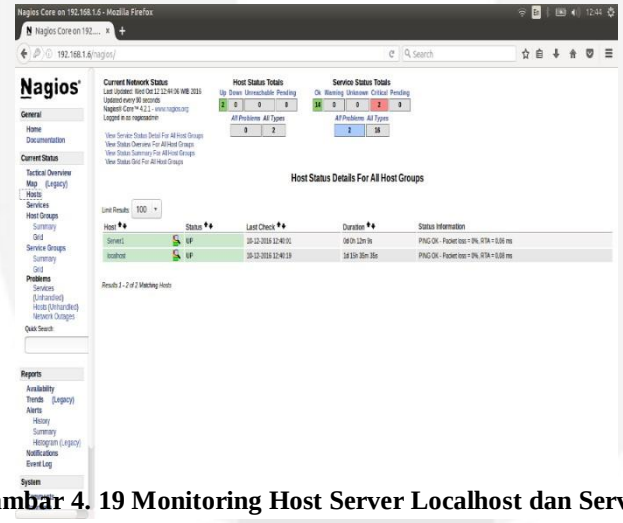
Gambar 4. 18 Notifikasi Email Nagios

Keterangan :

Waktu dan tanggal pada *email* menunjukkan waktu dan tanggal saat *Nagios* mengirim *notifikasi*, yaitu saat terjadi *error* pada *server*. Apabila terjadi *delay* pengiriman akibat gangguan pada *SMTP* dari *ISP*, tidak berpengaruh terhadap waktu dan tanggal pengiriman *notifikasi* oleh *Nagios*. Waktu dan tanggal sesuai dengan saat pengiriman meskipun diterima pada waktu yang berbeda.

4.10.2. Pengujian Monitoring Host dan Service

Pengujian dilakukan dengan cara melakukan akses ke *web interface server Nagios http://192.168.1.6/nagios*. Dibawah ini adalah gambar *web interface* dari *Nagios* :

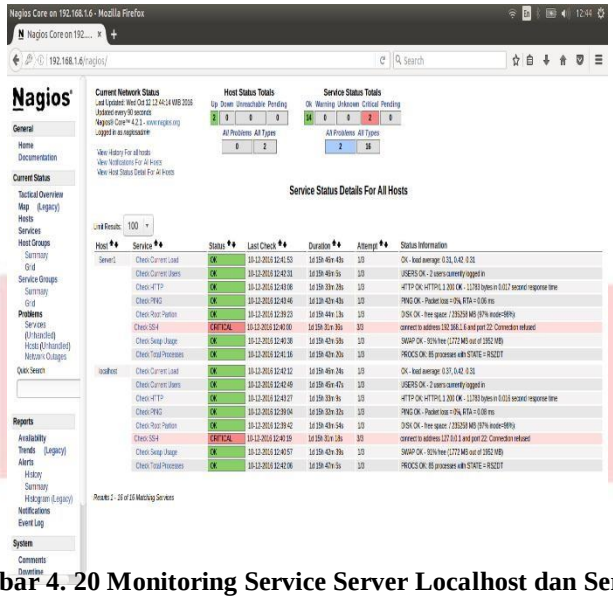


Gambar 4. 19 Monitoring Host Server Localhost dan Server1

Keterangan Gambar :

a. *Host* : menunjukkan jumlah *server* yang dimonitoring beserta status *servernya* dalam keadaan up atau *down*. Status *up* bila *server* hidup dan terkoneksi ke jaringan, status *down* bila *server* mati atau tidak terkoneksi ke jaringan.
b. *Services* : menunjukkan status dari service yang di *monitoring*. Terdapat beberapa status yaitu *Critical*, *Warning*, *Unknown*, *Ok*, dan *Pending*.
c. *Monitoring Features* : memperlihatkan fitur-fitur pada *Nagios* yang diaktifkan. Fitur tersebut diantaranya adalah *Flap Detection*, *Notification*, *Event Handler*, *Active Checks*, dan *Passive Checks*. Fitur yang berstatus *enable* berarti sedang aktif dan *disable* berarti tidak aktif

Dibawah ini adalah gambar *interface* yang menunjukan status *service* dari *host server* yang *dimonitoring* maupun pada *localhost*, sesuai pada konfigurasi yang dilakukan. *Service* yang tampil pada kolom *service* juga merupakan hasil konfigurasi.



Gambar 4. 20 Monitoring Service Server Localhost dan Server1

Keterangan :

Pada gambar terdapat bagian „*service status details for all hosts'* yang terdiri dari 7 kolom, mempunyai fungsi yang berbeda-beda :

Host :menunjukkan nama *host*

Service : menunjukkan *service* dari masing-masing host

Status : menunjukkan status dari *service*.

Last Check : menunjukkan waktu terakhir service dicek

Duration : lamanya service di *check*

Attempt : percobaan *check* yang dilakukan pada *service*

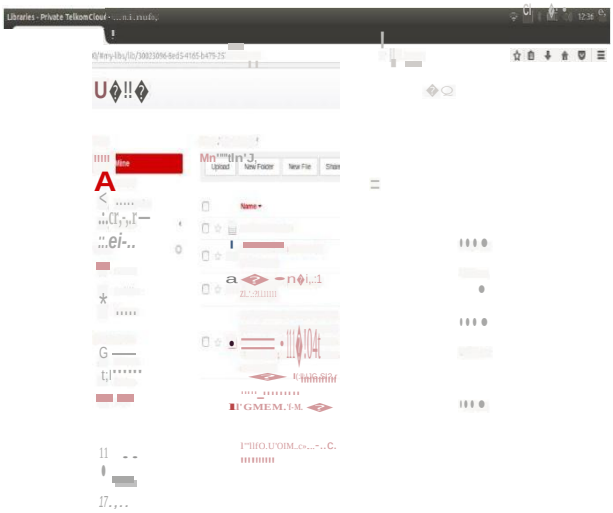
Status Information :memberikan informasi detail gangguan yang terjadi pada service.

Catatan : dari hasil penelitian ketika terjadi error terdapat *delay* pada perubahan status *service* dari "*OK*" ke "*CRITICAL*" maupun ke status lainnya dan sebalikanya. *Delay* terjadi selama 2-4 menit. Lamanya *delay* terjadi tergantung pada pendefinisian pada konfigurasi "*normal_check_interval*" pada *Nagios* dan pada auto *refreshwebbrowser*. Sama halnya yang terjadi pada perubahan status *host* dari "*UP*" ke "*DOWN*" maupun sebalikanya.

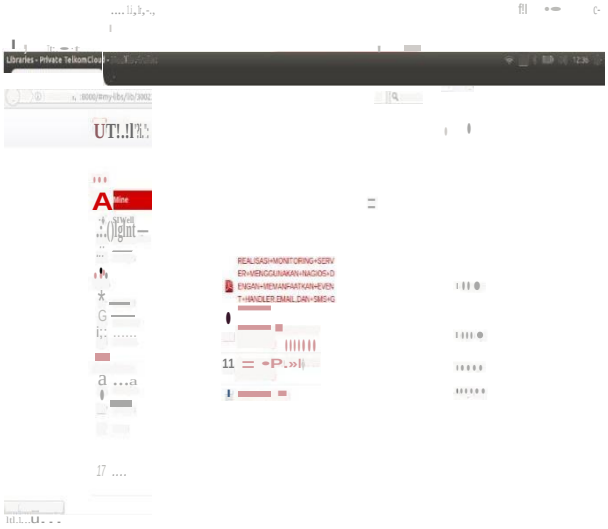
4.10.3. Pengujian Cloud Storage Seafile

4.10.3.1. Pengujian Penyimpanan Berkas Agar Terstruktur

Pengujian dilakukan dengan menggunakan fitur "*short by name*" atau "*short by size*" yang sudah tersedia di dalam aplikasi *Seafile* tanpa perlu mengkonfigurasinya.



Gambar 4. 21 Gambar Berkas yang diurutkan dari Z ke A



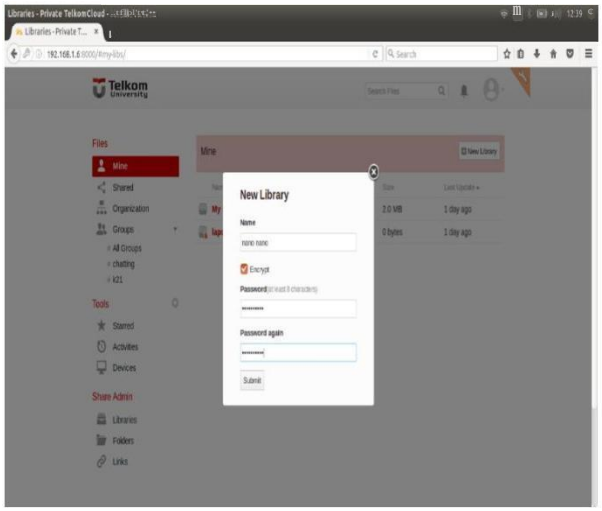
Gambar 4. 22 Gambar Berkas yang diurutkan dari A ke Z

Keterangan :

Gambar 4.21 mengurutkan berkas dari Z ke A menggunakan fitur "*Short by Name*" dan Gambar 4.22 mengurutkan berkas dari A ke Z menggunakan fitur "*Short by Name*".

4.10.3.2. Pengujian Keamanan Library

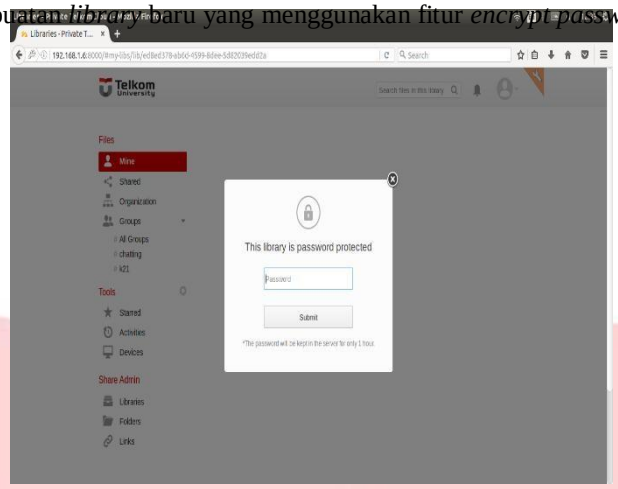
Pengujian dilakukan dengan memberikan *encrypt password* ketika pembuatan *library*, panjang *password* minimal sekitar 8 character tidak boleh kurang dari itu, seperti gambar di bawah ini :



Gambar 4. 23 Pembuatan Library yang di encrypt

Keterangan :

Pada gambar ini adalah pembuatan *library* baru yang menggunakan fitur *encrypt password*



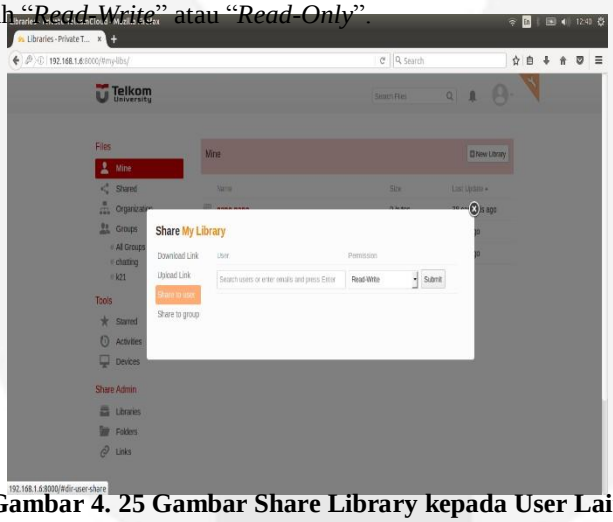
Gambar 4. 24 Gambar Ketika User Lain Membuka Library yang sudah di Encrypt

Keterangan :

Gambar ini akan muncul ketika user mencoba membuka *library* yang sudah di *encrypt*

4.10.3.3. Pengujian Hak Akses Library dan File

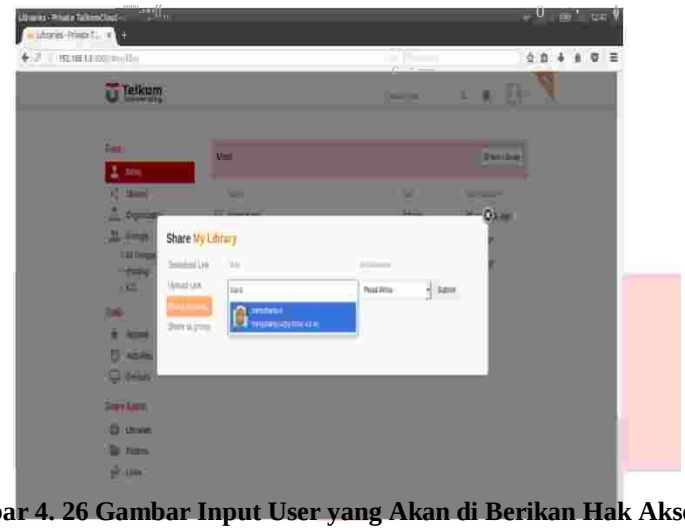
Pengujian ini mencoba *sharing library* dengan nama "*My Library*" dari *user* „cobasaja600" ke „tomybaraya", dengan cara mengklik *share* pada *library* dan kemudian Pilih "*Share to User*", Lalu ketikan nama *user* dan hak akses dari *user* tersebut apakah "*Read-Write*" atau "*Read-Only*".



Gambar 4. 25 Gambar Share Library kepada User Lain

Keterangan :

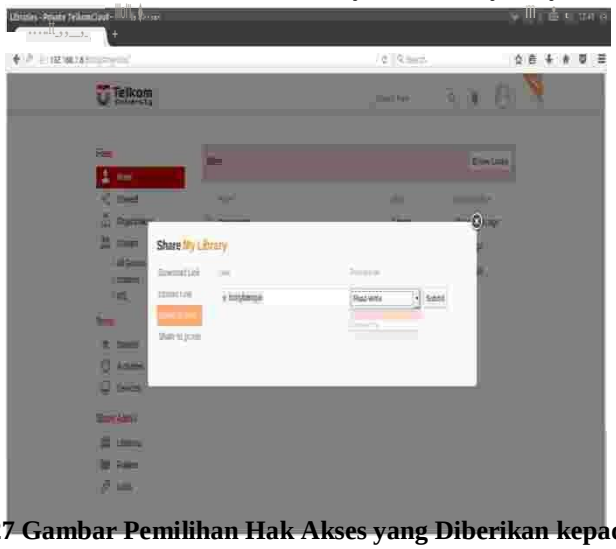
Gambar memperlihatkan cara *share library* ke *user* lain dengan menggunakan *menu share* yang ada pada *cloud storage Seafile*



Gambar 4. 26 Gambar Input User yang Akan di Berikan Hak Akses

Keterangan :

Mengiputkan nama *user* yang akan diberikan hak akses misalnya *user* "tomybaraya".

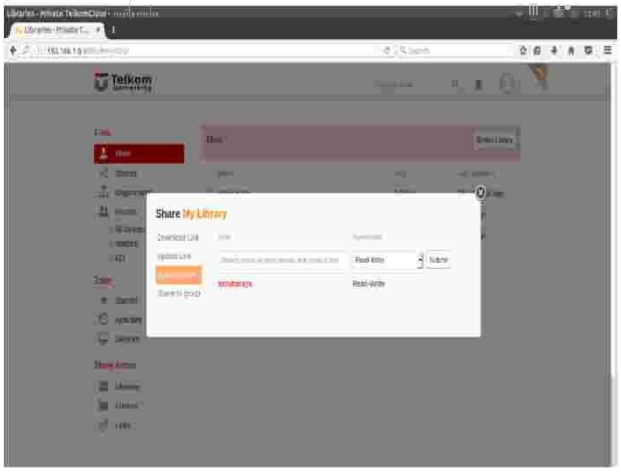


Gambar 4. 27 Gambar Pemilihan Hak Akses yang Diberikan kepada User Lain

Keterangan :

Pemberian hak akses kepada *user* yang sudah dipilih, hak akses tersebut antara lain :

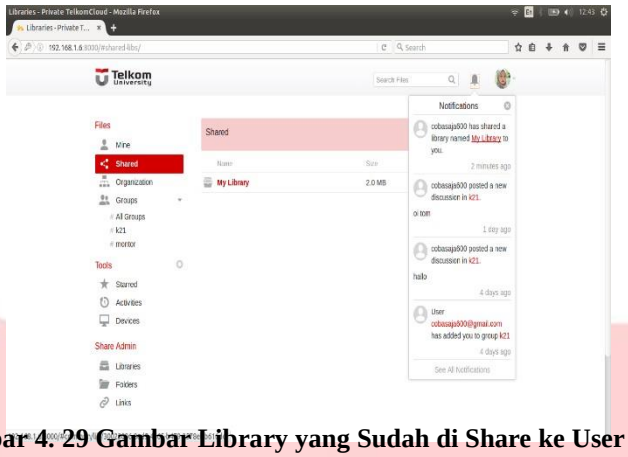
1. *Read-Write* : *user* yang diberikan hak akses bisa melihat dan merubah isi dari *library* yang sudah di *share*
2. *Read-Only* : *user* diberikan hak akses hanya untuk melihat isi dari *library* yang sudah di *share*



Gambar 4. 28 Gambar Ketika Sudah Memberikan Hak Akses kepada User Lain

Keterangan :

Gambar ini untuk melihat siapa saja *user* yang sudah diberikan hak akses dan dapat menghapus hak akses user pada *library* tersebut.



Gambar 4. 29 Gambar Library yang Sudah di Share ke User Lain

Keterangan :

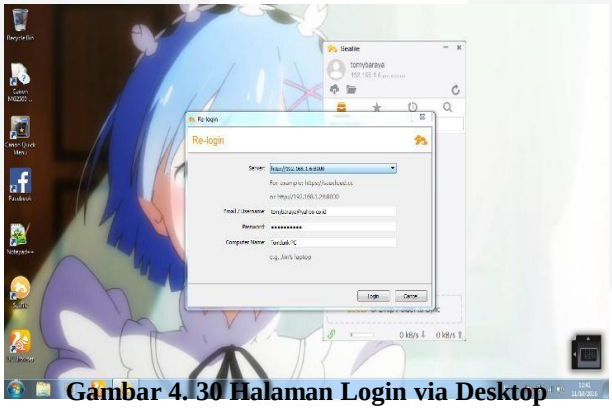
Gambar pengecekan ada atau tidaknya *library* yang sudah di *share* dan diberikan hak akses kepada *user* bersangkutan.

Catatan :

User lain tidak dapat melihat *Library* dari *user* lain kecuali user melakukan *sharing Library*.

4.10.3.4. Pengujian Hak Akses Library dan File

Pengujian mengakses *server cloud storage Seafile* menggunakan aplikasi *desktop Seafile* yang di install pada *Windows 7*.



Gambar 4. 30 Halaman Login via Desktop

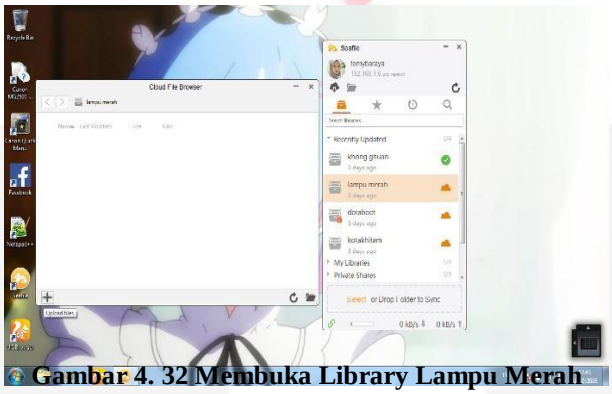
Keterangan :

Pada gambar ini perlu menginputkan *server http://192.168.1.6:8000*, Kemudian inputkan *email* dan *password user* yang sudah terdaftar di *cloud storage Seafile*.

Gambar 4. 31 Halaman Utama Menampilkan Recently Update

Keterangan :

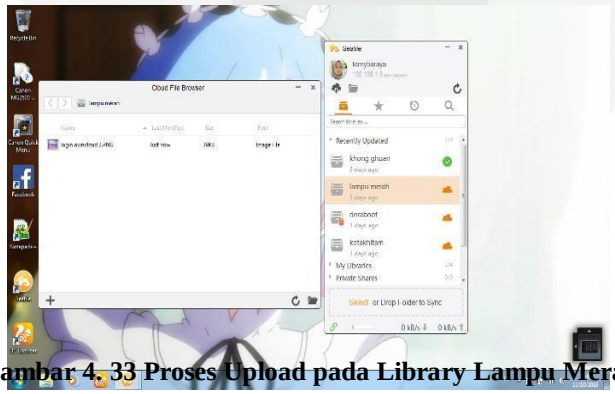
Pada gambar ini menampilkan *menu* "*Recently Update*" dan juga menampilkan halaman awal dari *Interface Seafile* versi *Desktop*.



Gambar 4. 32 Membuka Library Lampu Merah

Keterangan :

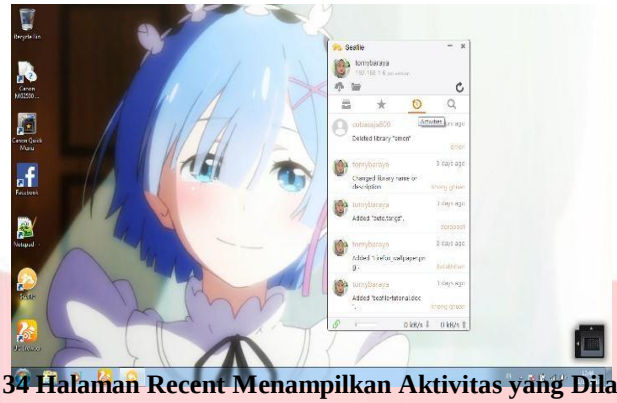
Pada gambar ini membuka dan melihat isi dari *library* "Lampu Merah".



Gambar 4. 33 Proses Upload pada Library Lampu Merah

Keterangan :

Gambar ini *user* mencoba melakukan proses *upload file* ke dalam *library* "Lampu Merah".



Gambar 4. 34 Halaman Recent Menampilkan Aktivitas yang Dilakukan User

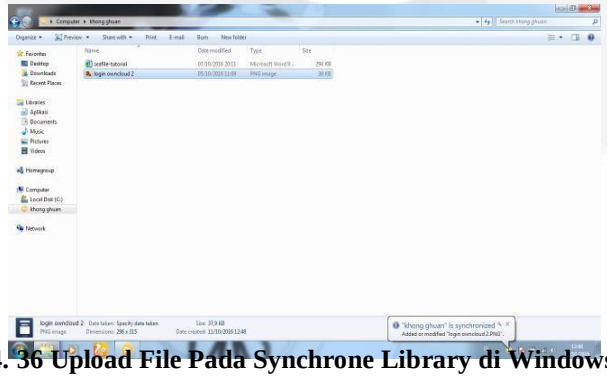
Keterangan :

Pada gambar ini menampilkan hasil aktivitas apa saja yang sudah di lakukan dari awal *user* mendaftar

Gambar 4. 35 Halaman Search Digunakan Untuk Searching File

Keterangan :

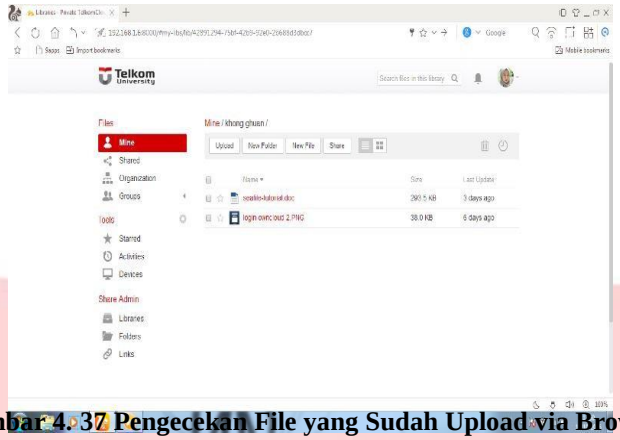
Pada gambar ini *user* dapat *searching file* yang ada di *library* dengan menggunakan menu *search* yang ada pada aplikasi



Gambar 4. 36 Upload File Pada Synchrone Library di Windows Explorer

Keterangan :

Gambar ini memperlihatkan proses *upload file* melalui *windows explorer* ke dalam *library* “Kong guan”.



Gambar 4. 37 Pengecekan File yang Sudah Upload via Browser

Keterangan :

Gambar pengecekan *file* yang sudah di *upload* dari *windows explorer* ke *library* "kong guan".

5. Kesimpulan

5.1. Kesimpulan

Berdasarkan analisis terhadap aplikasi serta evaluasi yang dilakukan sebelumnya, maka diperoleh kesimpulan sebagai berikut :

1. Dengan adanya fitur "*Sort by Name*" pada *cloud storage Seafile file-file* yang di-*upload* jadi lebih terstruktur.
2. Keamanan *cloud storage Seafile* dilengkapi dengan enkripsi *password* pada setiap masing-masing *library* meskipun *library* tersebut sudah di-*share*, sehingga *file* yang ada pada *library* terserbut hanya dapat diakses oleh *user* itu sendiri.
3. Terdapat 2 jenis hak akses, "*Read-Only*" yaitu *user* hanya bisa mengakses *file* dan tidak dapat menghapus *file* yang ada pada *library* terserbut, dan "*Read-Write*" yaitu *user* dapat mengakses dan menghapus *file* yang ada pada *library* tersebut.
4. Nagios dapat digunakan sebagai pemecahan masalah-masalah gangguan pada server dengan cara memonitoring *service-service* dan *host-host* pada komputer server.
5. Notifikasi *email* akan dikirim oleh *email* yang sudah didaftarkan pada Nagios apabila ada masalah pada *service* ataupun *host* pada komputer server.

5.2. Saran

Berikut adalah saran yang menjadi masukkan dalam pengembangan aplikasi.

1. Dapat ditambahkan sistem operasi server yang dimonitor
2. Dapat ditambahkan notifikasi sms yang ada pada Nagios.
3. Menambahkan fitur beberapa fitur pada Seafile agar bisa memainkan file audio maupun video didalam websitenya.

DAFTAR PUSTAKA

[1] Judith, H., Marcia, K., Fern, H., & Robin, B. (2010). *Cloud Computing for Dummies*. Indiana: Wiley Publishing.
[2] Purbo, O. W. (2012). *Membuat Sendiri Cloud Computing server menggunakan Open Source*. Yogyakarta: Penerbit Andi.
[3] Seafile. (2014). *Seafile Server Manual*. [Online]. Tersedia: http://freeplant.gitbooks.io/Seafile-server-manual/
[4] Nonprofit Technology Collaboration. (2013). *Cloud Storage*. [Online]. Tersedia: http://www.baylor.edu/business/mis/nonprofits/doc.php/197132.pdf
[5] Roebuck, K. (2011). *Cloud Storage: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Emero Pty Limited.

[6] Schulz,Greg. (2012). *Cloud and Virtual Data Networking*. Aurbach book.
[7] Galstad, E. (2008). *Nagios System and Network Monitoring. Nagios System and Network Monitoring*, 2.
[8] B, I. Y. (2010). *Flowchart, Algoritma, dan Pemrograman*. Yogyakarta: Graha Ilmu.
[9] Indonesia, W. F. (2009, November 3). Retrieved Maret 13, 2013, from worldfriend.web.id: http://www.worldfriend.web.id/pengertian-web-server
[10] Ubuntu Team. (2010-2014). *Getting Started with Ubuntu 14.04*. [Online]. Tersedia: http://ubuntu-manual.org/downloads
[11] Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.