

## IMPLEMENTASI LAYANAN PRIVATE CLOUD STORAGE MENGGUNAKAN OWNCLOUD DAN MONITORING DENGAN ZENOSS

### IMPLEMENTATION OF PRIVATE CLOUD STORAGE USING OWNCLOUD AND MONITORING WITH ZENOSS

Tomy Hidayat<sup>1</sup>, Fitri Susanti<sup>2</sup>, Yahdi Siradj<sup>3</sup>

<sup>1,2,3</sup>Prodi D3 Teknik Komputer, Fakultas Ilmu Terapan, Universitas Telkom

<sup>1</sup>[tomy0294@gmail.com](mailto:tomy0294@gmail.com), <sup>2</sup>[fitri.susanti@tass.telkomuniversity.ac.id](mailto:fitri.susanti@tass.telkomuniversity.ac.id), <sup>3</sup>[yahdi@tass.telkomuniversity.ac.id](mailto:yahdi@tass.telkomuniversity.ac.id)

---

#### Abstrak

Penyimpanan data menjadi salah satu kebutuhan bagi perusahaan kecil maupun perusahaan besar. Beberapa hal yang termasuk kebutuhan akan *storage* tersebut adalah menyimpan data atau file dokumen, berbagi file dan sinkronisasi. Namun hingga saat ini layanan penyimpanan data masih diimplementasikan pada jaringan lokal yang menggunakan teknologi LAN *sharing*. Manajemen file untuk berbagi file masih tidak terpusat pada *server* di teknologi LAN *sharing*. Akibatnya, data tersebut memiliki banyak kendala dalam hal *backup*, keamanan file, dan *sharing* file. Masalah tersebut bisa diatasi dengan owncloud karena dapat memajemen file terpusat pada satu *server*. Cloud tersebut dapat melakukan sinkronisasi file, sinkronisasi kontak dan sinkronisasi kalender. Fitur sinkronisasi file berfungsi untuk menyinkronkan file yang berada di layanan cloud menuju komputer lokal secara otomatis dan dapat diatur sesuai kebutuhan kita. Fitur *sharing* kontak berfungsi secara otomatis untuk menyimpan dan berbagi kontak antar *user*. Aplikasi kalender menolong *user* untuk membuat penanda kegiatan dari antar *user* hingga satu grup menjadi lebih mudah. Selain itu, sistem *monitoring* pada aplikasi zenoss dipasang untuk memantau performansi pada *PC server* khususnya penggunaan *memory* dan proses CPU.

**Kata kunci:** cloud computing, monitoring, sinkronisasi.

---

#### Abstract

*Data storage become one of necessity for little company as well as big company. Several things in storage necessity is storing data or document file, file sharing and synchronizing. However, until now that service still implemented on local network where still using LAN sharing technology. In other word, the data has problem for backup, file security and file sharing. Owncloud can solving that problem because can managing centralized file in one server. That cloud storage also can file synchronizing, contact synchronizing and calendar synchronizing. File synchronizing feature has function for synchronizing file in cloud service to local file automatically and can be arranged as our need. Contact synchronizing feature has function automatically for saving and sharing contact among user. Calendar synchronizing feature helping user for make agenda from among user until one group become easier. Beside that, monitoring system with application zenoss installed for looking performance on server PC especially monitoring memory usage and CPU process.*

**Keywords:** cloud computing, monitoring, synchronization.

---

## 1. Pendahuluan

Untuk beberapa perusahaan kecil maupun besar layanan *storage* hingga saat ini dibutuhkan, diantaranya untuk kebutuhan berbagi file dan sinkronisasi data yang ada. Namun hingga saat ini layanan penyimpanan data masih diimplementasikan pada jaringan lokal yang menggunakan teknologi LAN *sharing*, yaitu teknologi terhubung antara satu komputer dengan komputer lainnya. Manajemen file untuk berbagi file masih tidak terpusat pada teknologi LAN *sharing*. Karena tidak terpusat, penyimpanan data atau file dokumen tersimpan di banyak komputer. Akibatnya, data tersebut membutuhkan adanya manajemen dalam distribusi file. Cloud computing menjawab keterbatasan tersebut. Dengan adanya file yang terpusat pada *server*, maka masalah tersebut dapat diatasi dengan mudah. Sistem *monitoring* juga diperlukan untuk memantau kinerja dari *server*. Sistem tersebut berguna untuk memantau penggunaan memori dan proses CPU *server*.

*Cloud computing* memiliki banyak penggunaannya dan diantaranya adalah *cloud storage* dan *monitoring*. Owncloud merupakan *cloud storage* yang dapat mensinkronisasi file, *sharing* file, *upload* file dan *download* file. Aplikasi tersebut juga memiliki fitur sinkronisasi file, sinkronisasi kontak, dan sinkronisasi kalender. Fitur tersebut berguna sebagai pendukung aktivitas dari anggota perusahaan satu dengan lainnya. Selain bersifat *open source*, owncloud juga memiliki kelebihan seperti tidak dibatasi dalam akses user, cepat dalam sinkron koneksi internal, dan *sharing* pada aplikasi owncloud [1]. *Cloud storage* tersebut menyediakan pilihan dalam menggunakan sistem penyimpanan. Dengan menggunakan sistem operasi berbasis linux, sistem penyimpanan yang disediakan diantaranya SAN, NAS, langsung terpasang dan virtual, sesuai kebutuhan [2]. Sistem *monitoring* berguna untuk memantau penggunaan *memory usage* dan proses CPU. Salah satu aplikasi layanan monitoring adalah Zenoss. Zenoss merupakan aplikasi yang menyediakan *monitoring device* (switch, router, dll), proses, *service*, *file system* dan *routing*. Semua aktivitas perangkat dapat ditampilkan berupa grafik.

## 2. Dasar Teori

### 2.1 Cloud Computing

*Cloud Computing* adalah sebuah model komputasi yang membuat *resource Information and Technology (IT)* seperti banyak *server*, *middleware*, dan aplikasi mudah di akses melalui internet

kapanun sebagai layanan yang dapat digunakan secara umum atau *private* [3].

### 2.2 Cloud Storage

Cloud storage adalah sebuah layanan penyimpanan data online via internet atau intranet. Provider menyediakan layanan upload file pengguna ke dalam server eksternal. Pengguna senantiasa mem-backup data pada ketika menggunakan layanan cloud dikarenakan proses recover data membutuhkan waktu yang lebih lama dibandingkan backup lokal.[4]

### 2.3 Owncloud

OwnCloud adalah sebuah *enterprise file sync* dan dapat di hosting menggunakan *data center* pengguna, *server* pengguna, dan menggunakan *storage* pengguna. Pengguna bisa mengakses file perusahaan melalui berbagai perangkat, kapanpun, dimanapun disaat IT dapat dikelola, dikendalikan dan mengaudit *file sharing activity* untuk memastikan keamanan dan merapikan data sesuai ukurannya. [5]

### 2.4 Private Cloud dan Public Cloud

*Private cloud* adalah *cloud* yang dibuat hanya untuk lingkungan internal. *Server* dari *cloud* tersebut termasuk digunakan di dalam organisasi. Jika *private cloud* dibangun oleh pihak lain/ lingkungan luar dan *server* tersebut berada diluar organisasi, maka *private cloud* tersebut disebut *virtual private cloud*. *Public cloud* merupakan *cloud* yang dapat diakses oleh sembarang pengguna, dan melalui jaringan internet. [6]

### 2.5 Sinkronisasi

Suatu proses yang bekerja bersama-sama dan saling berbagi data dapat mengakibatkan *race condition* atau pengaksesan data secara bersama-sama[7]

### 2.6 Monitoring

*Monitoring* adalah sistem yang digunakan untuk mendeteksi apakah sistem dan jaringan komputer dalam keadaan layak untuk digunakan. Keadaan layak pada jaringan komputer dalam berkomunikasi seperti ping *server* lancar, utilisasi CPU bekerja normal/ stabil, suhu komputer yang normal (rata-rata dibawah 50 derajat dengan processor yang berbeda). Hasil dari monitoring tersebut dapat membantu admin dalam mendesain ulang sebuah jaringan. Hal yang sering dimonitoring beberapa diantaranya seperti utilisasi CPU, suhu komputer, ping *server*, dan HTTP.[8]

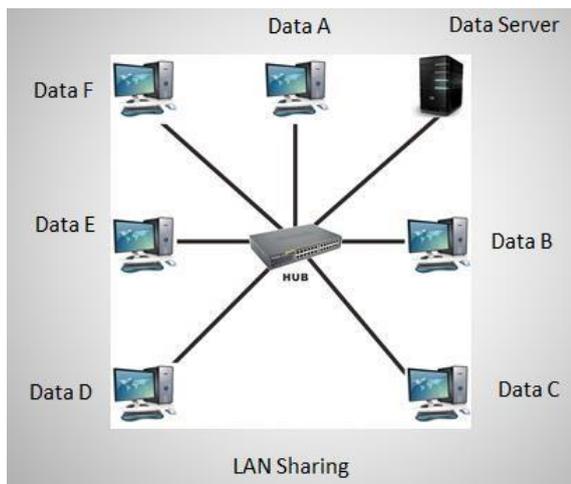
**2.7 Zenoss**

Zenoss core adalah aplikasi berbasis *web* yang dapat menginstallasi *server* pusat pada jaringan dan menggunakan aplikasi *server* zope. Aplikasi tersebut diprogram menggunakan bahasa Python. Aplikasi tersebut berbasis linux, tetapi tidak mewajibkan menjadi linux *administrator* untuk menginstallasi dan menggunakan zenoss core.[9]

**3. Analisis dan Perancangan**

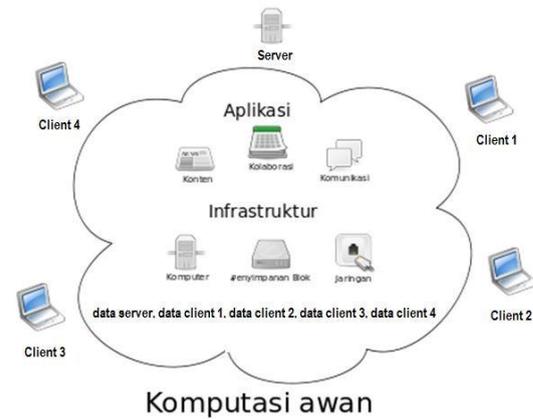
**3.1 Gambaran Sistem Saat Ini**

Berikut gambaran sistem sebelum dilakukan implementasi ownCloud. *Server* dan semua *client* terhubung melalui hub sehingga membentuk LAN *sharing*. *Data sharing* yang digunakan masih tersebar antara *server* dan *client*. Dibutuhkan waktu yang lama jika salah satu *client* atau *server* membutuhkan banyak data *sharing* yang terdapat lebih dari satu komputer.



**Gambar 3.1 Penggunaan LAN Sharing**

Jika dibandingkan dengan implementasi ownCloud, sistem digambarkan seperti berikut.

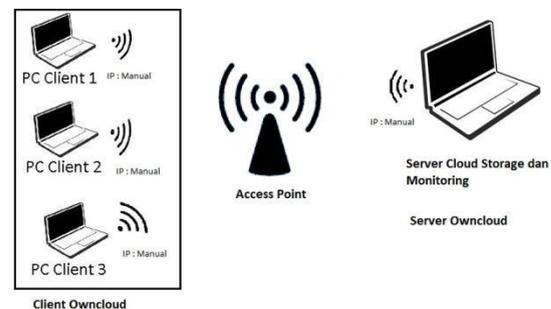


**Gambar 3.2 Gambaran Cloud Computing**

Dengan adanya teknologi komputasi awan atau biasa disebut *cloud*. Penyimpanan data tersebut bisa terkumpul atau terpusat pada satu *server*. Dengan terpusatnya data pada *server*, maka aktivitas *sharing* yang dibutuhkan untuk data yang banyak menjadi lebih cepat.

**3.2 Analisis Kebutuhan Sistem**

Tujuan dari pembuatan proyek akhir ini adalah membangun layanan *cloud storage* dan memonitori layanan *cloud* tersebut. Sebagai gambaran dari proyek tersebut, maka dibuatlah desain implementasi seperti gambar dibawah

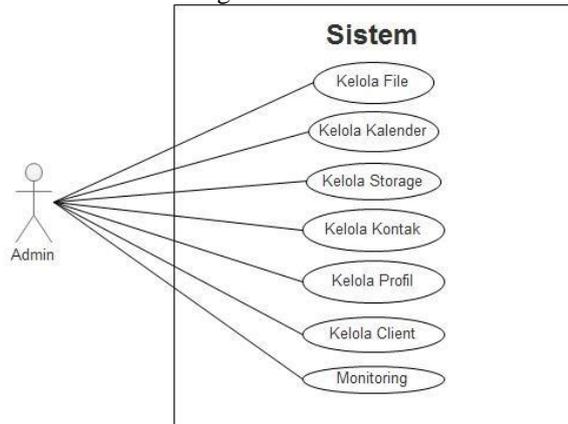


**Gambar 3.3 Desain Private Cloud Storage**

Dari gambar diatas *cloud storage* yang dibangun terdiri dari 1 buah *server* dengan layanan *cloud storage* dan minimal 1 buah *client*. *Server* tersebut dapat melakukan *monitoring host* menggunakan aplikasi zenoss dan menyediakan layanan dari *owncloud*. Yang dilakukan *monitoring* oleh *server* adalah utilitas CPU dan penggunaan memori. Layanan *server* yang diberikan dari *owncloud* salah satunya adalah *cloud storage*. *Cloud storage* tersebut memberikan layanan simpan data untuk *client* via lokal. IP yang disediakan *server private* adalah manual karena masih belum bisa diimplementasikan sehingga tidak tetap dan statis.

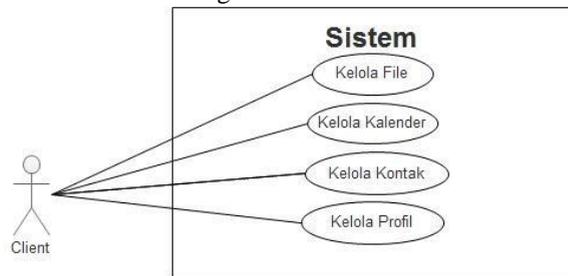
### 3.3 Use Case

Berikut use case diagram untuk admin



Gambar 3.4 Use Case Diagram Cloud Storage Admin

Berikut use case diagram untuk client



Gambar 3.5 Use Case Diagram Cloud Storage Client

Deskripsi use case

Tabel 3.1 Deskripsi Use Case Diagram

Use Case	User	Deskripsi Fungsionalitas
Kelola File	Admin dan Client	Dapat mengupload file
	Admin dan Client	Dapat mendownload file
	Admin dan Client	Dapat membuat folder
	Admin dan Client	Dapat membuat text file
	Admin dan Client	Dapat shared folder
Kelola Kalender	Admin dan Client	Dapat menambahkan kategori event
	Admin dan Client	Dapat mengubah kategori event
	Admin dan Client	Dapat menghapus kategori event
	Admin dan Client	Dapat menambahkan detail kategori event
	Admin dan Client	Dapat mengubah detail kategori event
	Admin dan Client	Dapat mengubah warna kategori event
	Admin dan Client	Dapat share kategori event
	Admin dan Client	Dapat menambahkan detail share event
	Admin dan Client	Dapat mengubah detail share event
	Admin dan Client	Dapat menghapus detail share event

	Admin dan Client	Dapat menambahkan fitur <i>share event</i>
	Admin dan Client	Dapat mengubah fitur <i>share event</i>
Kelola Storage	Admin	Dapat mengubah ukuran quota penyimpanan client atau admin
Kelola Kontak	Admin dan Client	Dapat menambah kontak
	Admin dan Client	Dapat mengubah kontak
	Admin dan Client	Dapat menghapus kontak
Kelola Client	Admin	Dapat menambah admin atau client
	Admin	Dapat merubah tampilan nama admin atau client
	Admin	Dapat merubah password admin atau client
	Admin	Dapat merubah status user dari admin ke client atau sebaliknya
	Admin	Dapat merubah besar batas kapasitas storage
Kelola Profil	Admin dan Client	Dapat menampilkan penggunaan besar kapasitas storage
	Admin dan Client	Dapat merubah password
	Admin dan Client	Dapat merubah tampilan nama
Monitoring	Admin	Dapat melihat <i>memory usage</i>
	Admin	Dapat melihat proses CPU

### 3.4 Kebutuhan Perangkat Keras dan Perangkat Lunak

Adapun spesifikasi setiap perangkat komputer untuk pengembangan sistem adalah sebagai berikut :

Tabel 3.2 Spesifikasi Komputer Saat Ini

Kebutuhan Perangkat keras dan Perangkat lunak		
No.	PC Server	PC Client
1. Perangkat keras	RAM 4 GB	RAM 4 GB
	Processor Core i-3	Processor Core i-3
	Wifi	Wifi
	Harddisk sebesar 500 GB	Harddisk sebesar 50 GB
	2. Perangkat Lunak	Sistem Operasi Centos 6.5
	mySQL/MariaDB 5.5	-
	Apache 2.4	-
	PHP 5.6	-
	Mozilla Firefox 27	Mozilla Firefox 27

	Owncloud Server 8.2	Owncloud desktop Client
	Zenoss 4.25	-

Berdasarkan referensi [9] dan [13], bentuk spesifikasi perangkat keras dan perangkat lunak minimal adalah sebagai berikut untuk setiap PC adalah berikut :

Tabel 3.3 Spesifikasi Komputer minimal

Kebutuhan Perangkat Keras dan Perangkat Lunak		
No.	PC Server	PC Client
1. Perangkat keras	RAM 4GB	RAM 4GB
	Processor Intel Core 2	Processor Core i-3
	Wifi	Wifi
	Harddisk sebesar 100 GB	Harddisk sebesar 50 GB
2. Perangkat Lunak	Sistem Operasi Centos 6.5	Sistem Operasi Windows 7
	MySQL/MariaDB 5.5	-
	Apache 2.4	-
	PHP 5.4	-
	Mozilla Firefox 27	Mozilla Firefox atau Google Chrome
	Owncloud Server 8.2	Owncloud desktop Client
	Zenoss 4.25	-

## 4. Implementasi dan Pengujian

### 4.1 Implementasi

Implementasi pada sistem ini secara umum terbagi menjadi tiga, yaitu instalasi dan konfigurasi owncloud server, instalasi dan konfigurasi owncloud desktop, kemudian instalasi dan konfigurasi pada aplikasi monitoring zenoss.

## 1.2 Pengujian

### 1.2.1 Skenario Pengujian

Berikut skenario pengujian yang dilakukan sebagai berikut :

#### 1. Server

Server menggunakan alamat cloud dan monitoring sebagai berikut:

- Alamat cloud OwnCloud :  
http://localhost/owncloud
- Alamat monitoring zenoss :  
http://localhost:8080
- Server menggunakan aplikasi owncloud server versi 8.2 untuk pengujian server
- Server menggunakan aplikasi zenoss versi 4.2.5 untuk pengujian monitoring

#### 2. Client

Client menghubungkan ke server cloud menggunakan alamat  
http://192.168.43.171/owncloud

#### 3. Daftar user ownCloud

Berikut data user beserta grup yang ditempatkan untuk masing-masing user

Username	Full Name	Password	Groups
admin	admin	••••••••	admin
bejo	bejo	••••••••	client, les
udin	udin	••••••••	client, les

Gambar 4.37 Daftar User Owncloud

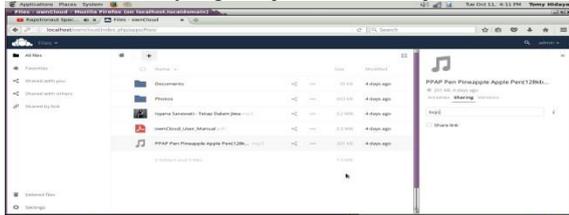
User berjumlah tiga untuk uji coba fitur ownCloud. Terdiri dari satu admin dengan username admin dengan grup admin dan dua user client dengan nama bejo dan udin. Untuk client bejo dan udin ditempatkan di group client dan les. Group akan digunakan pada pengujian sinkronisasi kalender.

- Untuk menguji sinkronisasi file dengan menguji fitur share yang ada di aplikasi owncloud antara server dan client. Jenis share yang dilakukan berupa share file menuju direktori user tujuan.
- Pengujian sinkronisasi kontak dilakukan di aplikasi kontak. Yang akan diuji dari aplikasi ini adalah cara kerja aplikasi tersebut. Fitur sinkronisasi dari aplikasi kontak adalah fitur share kontak.
- Untuk pengujian sinkronisasi kalender yang dijadikan parameter adalah fitur share event yang berada di aplikasi kalender. Fitur tersebut akan diuji beberapa diantaranya berupa tampil di kalender user tujuan dan dapat menampilkan informasi detail tentang event dari fitur tersebut.

### 1.2.2 Pengujian Sinkronisasi File

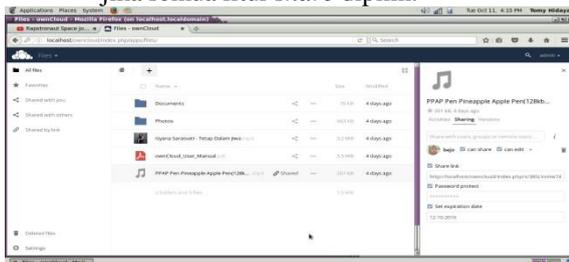
Pada pengujian ini akan dicoba sinkronisasi file antar *user*. Parameter yang digunakan adalah dengan beberapa cara *share* file antar *user*. Berikut langkah sinkronisasi file yang dilakukan :

1. Admin membuat file *share* menuju *client*. Client yang dituju bernama bejo.



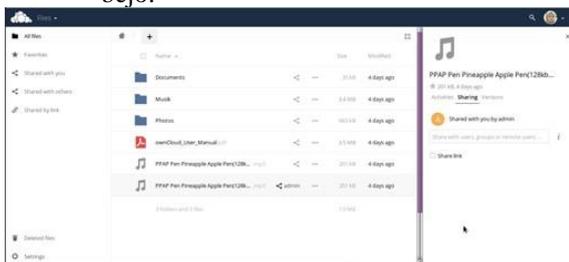
Gambar 4.38 Membuat Shared File

2. Admin membuat hak akses *share* file untuk bejo. Admin menambahkan hak akses file diantaranya dapat *menshare* ulang, dan dapat mengedit file. Kemudian admin menambahkan *share link*. *Link share* akan muncul otomatis dari owncloud. *Link share* tersebut dapat diatur dengan *password* dan tanggal kadaluwarsa dari *link* tersebut. Berikut gambar dibawah jika semua fitur *share* dipilih.



Gambar 4.39 Admin Membuat Hak Akses Share, Share Link, Enkripsi Password dan Tanggal Kadaluwarsa

3. *Client* bejo membuka file *share*. Gambar dibawah menunjukkan file *share* tersebut muncul di halaman kelola file milik *client* bejo.



Gambar 4.40 Client Membuat Shared File

Terlihat ikon *share* dengan nama admin, jika ikon *share* di klik muncul informasi bahwa file tersebut *dishare* oleh admin.

Berikut tampilan jika memilih menu *Shared with you* seperti gambar dibawah.

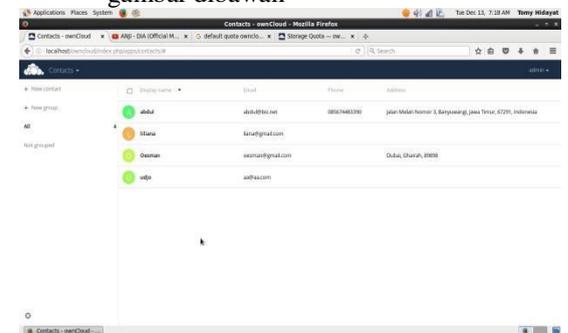


Gambar 4.41 Shared File Berada di Halaman Shared With You

### 4.2.3 Pengujian Sinronisasi Kontak

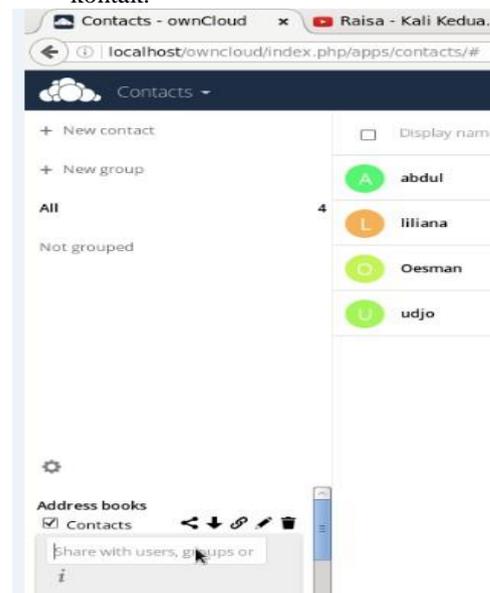
Pada pengujian ini akan dicoba bagaimana satu *user* dapat sinkronisasi daftar kontak dengan yang lain. Fitur yang digunakan untuk sinkronisasi adalah *sharing* kontak.

1. Admin membuat kontak bernama abdul, liliana, oesman dan udjo. Keempat kontak tersebut memiliki data lengkap seperti gambar dibawah



Gambar 4.42 Admin Membuat Kontak

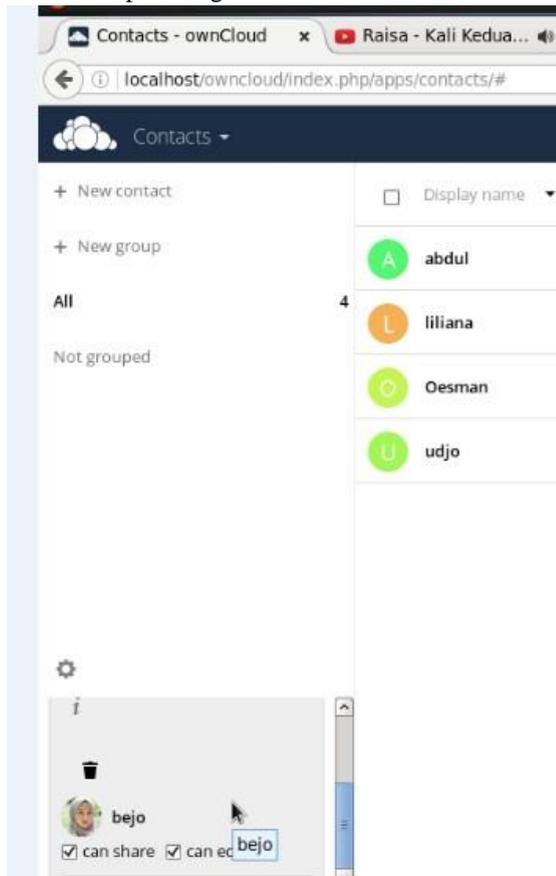
2. Admin *menshare* kontak ke kontak lain dengan menekan ikon *share* pada daftar kontak.



Gambar 4.43 Admin Share Kontak

3. Admin melakukan *share* kontak menuju user *client* bernama bejo. Ketentuan fitur *share* yang dapat dilakukan user bejo yaitu

dapat share ulang kontak tersebut dan dapat mengedit file.



Gambar 15.44 Share Kontak Menuju Bejo

4. Berikut ketika didalam aplikasi kontak user bejo, daftar client share berasal dari admin langsung ditampilkan di halaman kontak.



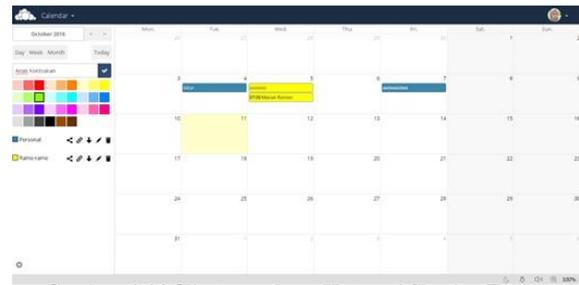
Gambar 4.45 Kontak Admin Berhasil Di-share

#### 4.2.4 Pengujian Sinkronisasi Kalender

Pada pengujian ini menggunakan fitur *share event*. Fitur ini bertujuan untuk memberi informasi ke *user* lain pada tanggal yang ditentukan *user* tersebut. Pengujian sinkronisasi dibagi menjadi dua, yaitu untuk *user* dan grup. Berikut langkah sinkronisasi kalender.

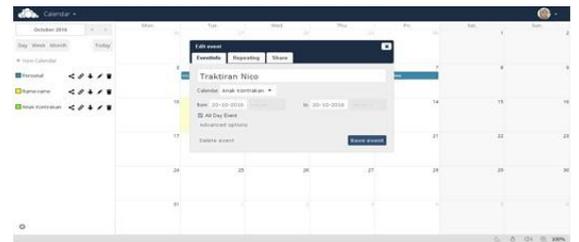
##### 4.2.4.1 Pengujian Sinkronisasi Kalender User

1. *Client* bejo membuat kategori kalender baru bernama Anak Kontrakan.



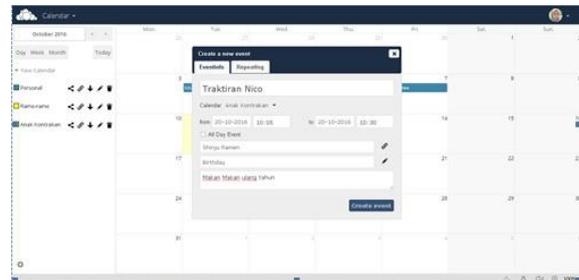
Gambar 4.46 Client membuat Kategori Sharing Event

2. Kemudian *client* bejo membuat *share event* di tanggal 20. Detil *event* tersebut berjudul Traktiran Nico dengan kategori kalender Anak Kontrakan.



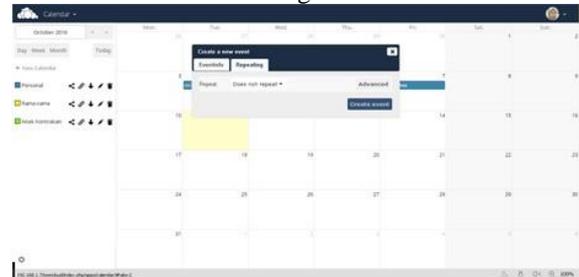
Gambar 4.47 Detil Share Event Kalender

3. Detil *event* diatur jam awal dan jam akhir dengan mengosongkan kotak *All Day Event*. Kemudian ditambahkan lokasi, jenis kegiatan dan keterangan dari *event* tersebut.



Gambar 4.48 Penambahan Detil Share Event Kalender

4. Karena kegiatan tersebut tidak dilakukan berulang maka tab *repeating* dipilih *does not repeat*. Kemudian event tersebut selesai dibuat dengan klik *Create Event*.



Gambar 4.49 Fitur Repeating di Kalender

5. Kalender selesai dibuat. Untuk memunculkan *event* di kalender diperlukan *refresh browser* satu kali.



Gambar 4.50 Event Kalender Berhasil Dibuat

6. Kemudian bejo melakukan *share* kalender menuju admin.



Gambar 4.51 Share Event Kalender

7. Tanda hasil *share* di admin seperti gambar dibawah



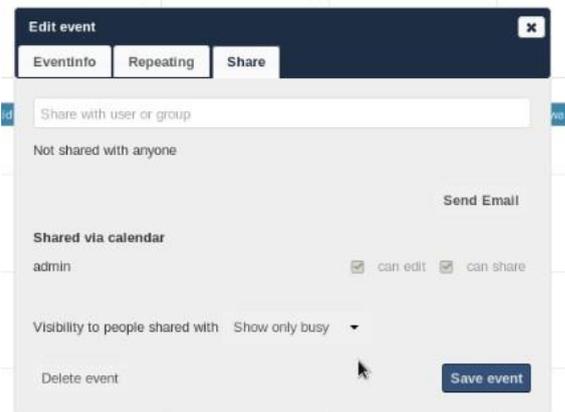
Gambar 4.52 Hasil Share Dari Bejo

8. Kemudian dari admin diperiksa hasil *share event* kalender. Di tanggal 20 muncul tanggal *share* dan tampil detail tanggal seperti gambar dibawah.



Gambar 4.53 Detil Share Kalender Dari Bejo

9. Di detail *event* muncul tab *share*. Tab ini menampilkan informasi akun yang telah mendapatkan *share* kalender. Kemudian selain itu muncul pilihan *visibility* atau tampilan *event* dari kalender tersebut. *Visibility* tersebut ada tiga pilihan yaitu all event, *show only busy*, dan *hide event*. Berikut penjelasan dari setiap *event* :  
*All event* : *event* tersebut akan selalu muncul di kalender *user* tujuan.  
*Show only busy* : akan muncul namun nama event diganti dengan nama busy.  
*Hide event* : *event* tersebut disembunyikan dari kalender *user* tujuan.



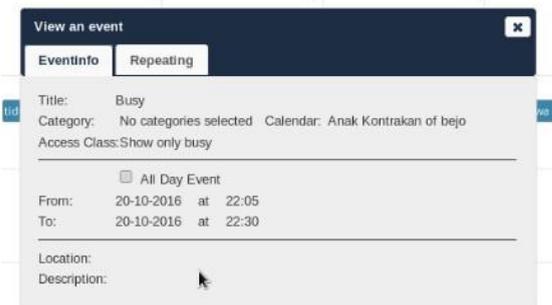
Gambar 4.54 Detil Event Share

10. Berikut tampilan jika *visibility* diubah dalam mode *show only busy*.



Gambar 4.55 Tampilan Show Only Busy

11. Jika *event busy* tersebut dibuka akan tampil detail sebagai berikut.



Gambar 4.56 Detil Event Busy

### 15.2.4.2 Pengujian Sinkronisasi Kalender Grup

Berikut pengujian sinkronisasi untuk grup.

1. Client Bejo membuat kegiatan LES PRIVAT dengan kategori kalender Bimbel. Kegiatan tersebut dilakukan pada tanggal 21-12-2016 jam 10.30 – 12.30.



Gambar 4.57 Bejo Membuat Kegiatan Bimbel

2. Kemudian kegiatan tersebut dishare menuju grup yang bernama les. Untuk mengetahui user yang ditempatkan di grup les, silahkan cek skenario pengujian nomor 3.



Gambar 4.58 Kegiatan LES PRIVAT di-share Menuju Grup Bernama les

- Berikut hasil share tersebut ditampilkan menuju anggota dari grup les. Udin merupakan salah satu anggota dari grup les. Terlihat keterangan “Dibagikan dengan anda dan group les oleh bejo”. Kemudian tampilan tanggal kegiatan telah terlihat di kalender user udin.



Gambar 4.59 Pengujian Share Kalender Berhasil

### 4.2.5 Pengujian Monitoring

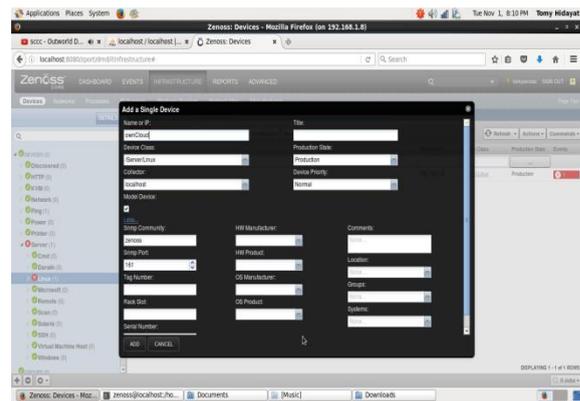
Pengujian monitoring memantau penggunaan memori dan penggunaan CPU dari komputer server. Untuk memantau komputer server terdiri dari beberapa langkah seperti berikut :

- Tambahkan device untuk komputer lokal. Karena yang akan dimonitoring komputer lokal, maka alamat yang dituju yaitu alamat localhost atau 127.0.0.1 atau 192.168.43.171. Masuk kedalam tab interface, kemudian pilih add single device.

Data yang dimasukkan untuk komputer lokal seperti berikut

- Nama atau IP : 127.0.0.1 atau ipv4 PC Server 192.168.43.171.
- Device Class : /Server/Linux
- Collector : localhost
- Title : ownCloud
- Device Priority : Normal
- Buka More, dan isi Snmp Community dengan nama zenoss dan snmp port 161
- Klik ADD.

Untuk lebih jelasnya ditampilkan gambar seperti dibawah



Gambar 4.60 Tambahkan Device Untuk Monitoring

- Edit konfigurasi snmp di /etc/snmp/snmpd.conf dengan kode

**nano /etc/snmp/snmpd.conf**

Dan tambahkan rocommunity public di file tersebut.

Untuk lebih jelasnya seperti gambar dibawah



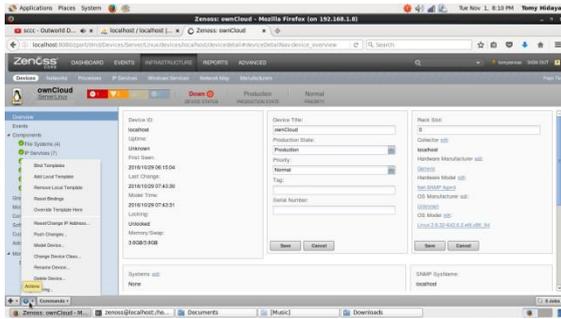
Gambar 4.61 Tambahkan rocommunity di snmp.conf

- Setelah mengkonfigurasi snmp, restart snmp dengan kode seperti dibawah

**service snmpd restart**

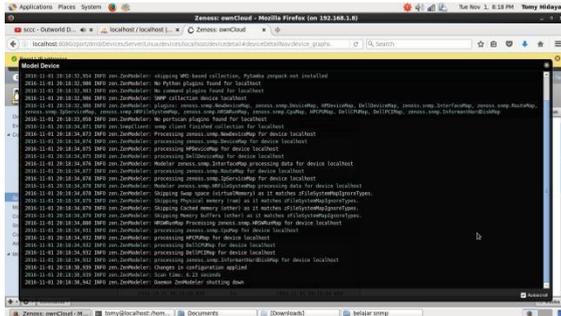
- Setelah restart, terlihat keterangan dari komputer lokal dari tab overview. Keterangan tersebut beberapa diantaranya seperti berikut :
  - Device ID : localhost
  - Uptime : Unknown
  - First Seen : 29/10/2016 pada jam 06:15:04
  - Last Change : 29/10/2016 pada jam 07:43:30
  - Model Time : 29/10/2016 pada jam 07:43:31
  - Locking : Unlocked
  - Memory/Swap : 3.6GB / 3.9GB
  - Device Title : ownCloud

Kemudian buka model device zenoss melalui tab interface, kemudian buka model device tersebut dengan ikon gear dibawah



Gambar 4.62 Buka Ikon Gear Untuk Model Device

Maka tampilan *model device* seperti gambar dibawah

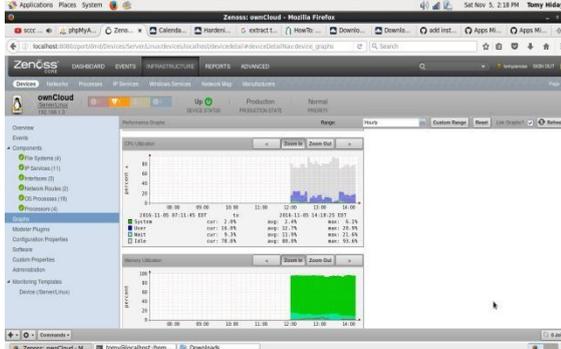


Gambar 4.62 Tampilan Model Device

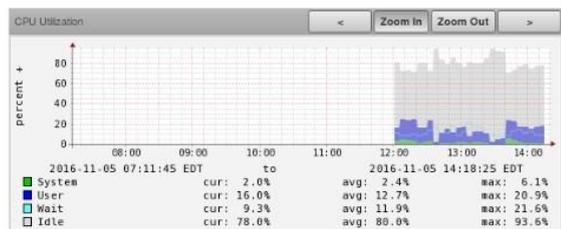
Kemudian tutup jendela model device setelah Daemon ZenModeler *shutting down*

5. Kemudian buka grafik di tab Graph, terlihat grafik dari penggunaan memori dan penggunaan CPU pada komputer *server*.

Gambar dibawah menunjukkan grafik dari penggunaan CPU



Gambar 4.63 Tampilan Grafik CPU Server



Gambar 4.64 Tampilan Grafik CPU Server detail

Dari gambar diatas terlihat persentase grafik dalam rentang waktu pada tanggal 05-11-2016 atau hari sabtu dari jam 08.00 menuju 14.00. Kemudian

diberikan detil grafik komputer dari jam 12.00 menuju 14.00. Dari grafik tersebut diberikan detil keterangan seperti tabel dibawah

Tabel 15.1 Detil Keterangan Grafik CPU Server

Nama	Warna	Status Grafik		
		Currency (terakhir)	Average (rata-rata)	Maximum (maksimal)
System	Hijau	2.0%	2.4%	6.1%
User	Biru Tua	16.0%	12.7%	20.9%
Wait	Biru Muda	9.3%	11.9%	21.6%
Idle	Abu-abu	78.0%	80.0%	93.6%

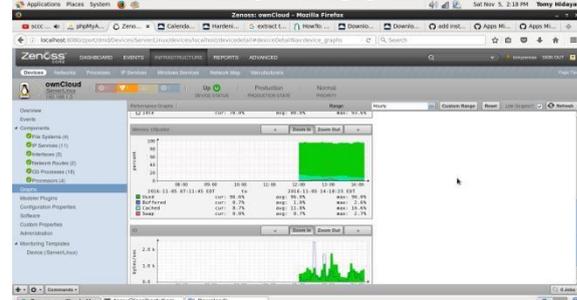
Penjelasan :

**System** merupakan prosentase kemampuan CPU yang digunakan saat ini oleh sistem. Yang digunakan oleh sistem adalah proses bawaan dari sistem komputer *server* yaitu centOS.

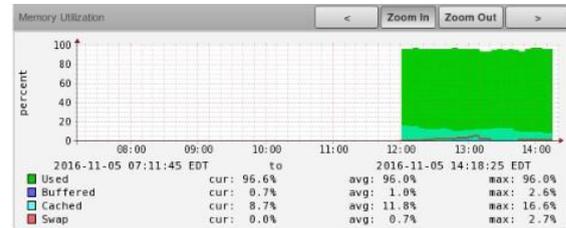
**User** merupakan prosentase kemampuan CPU yang digunakan saat ini oleh user. Yang digunakan oleh user berupa aplikasi yang dibuka oleh user. Misal browser mozilla firefox.

**Wait** merupakan proses kemampuan CPU menunggu respon dari disk

**Idle** merupakan prosentase kemampuan CPU yang diam tidak melakukan proses apapun.



Gambar 4.65 Tampilan Grafik Penggunaan Memori Server



Gambar 4.66 Tampilan Grafik Penggunaan Memori Server Detail

Dari gambar diatas terlihat persentase grafik dalam rentang waktu pada tanggal 05-11-2016 atau hari sabtu dari jam 08.00 menuju 14.00. Kemudian diberikan detil grafik komputer dari jam 12.00 menuju 14.00. Dari grafik tersebut diberikan detil keterangan seperti tabel dibawah

Tabel 15. 2 Tabel Detil Keterangan Grafik Penggunaan Memori Server

Nama	Warna	Status Grafik		
		Currency (terakhir)	Average (rata-rata)	Maximum (maksimal)
Used	Hijau	96.6%	96.0%	96.0%
Buffered	Biru Tua	0.7%	1.0%	2.6%
Cached	Biru Muda	8.7%	11.8%	16.6%
Swap	Merah	0.0%	0.7%	2.7%

Penjelasan .

**Used** merupakan prosentase besarnya memori yang sedang digunakan.

**Buffered** merupakan prosentase besarnya memori penyangga kerja perangkat keras dengan aplikasi yang digunakan. Misal kerja mouse bergerak dengan gerak aplikasi kursor mouse pada komputer, *burning cd* yang membutuhkan kerja *cd writer* dengan aplikasi *burning*.

**Cache** merupakan bagian memori yang digunakan untuk mempercepat suatu proses akses terhadap data. Memori tersebut merupakan sebuah tampungan data sementara untuk membuka ulang aplikasi secara cepat. Misal buka pertama kali aplikasi browser kemudian keluar. Kemudian browser tersebut dibuka kembali akan lebih cepat dari sebelumnya disebabkan cache memori tersebut. Pada aplikasi monitoring menampilkan prosentase besarnya tampungan yang digunakan.

**Swap** merupakan sebuah ruang virtual memory. Swap tersebut menjadi ruang memori tambahan jika kapasitas memory tidak mampu menampung data.

**4.2.6 Kasus Monitoring**

Berikut contoh beberapa kasus yang dapat merubah keadaan dari grafik cpu dan grafik memory:

Berikut tampilan grafik lima menit ketika *server* dinyalakan. Aplikasi user yang dijalankan hanya browser firefox untuk melihat grafik dari zenoss. Terminal yang terlihat hanya terbuka sementara dan tidak membuat perubahan besar pada grafik.



Gambar 4.67 Grafik Memori dan CPU Keadaan Awal Komputer

Gambar diatas menunjukkan beberapa keterangan seperti berikut :

Jam *server* menunjukkan 3:13 am.

Kemudian penjelasan grafik seperti tabel dibawah

Tabel 4. 3 Tabel Keterangan Grafik Keadaan Awal Komputer

Nama keterangan	Besar penggunaan terkini	Rata-rata
Used	93.1%	93.1%
Buffered	0.3%	0.3%
Cached	15.4%	15.4%
Swap	0.0%	0.2%

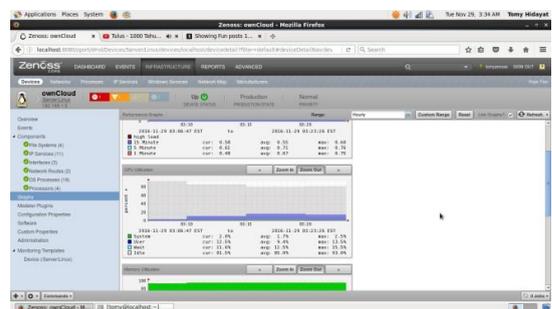
**Kasus 1**

Pada jam 3.21 am waktu *server*. *Server* menjalankan streaming youtube selama 5 menit. Berikut tampilan streaming.



Gambar 4.68 Tampilan Streaming

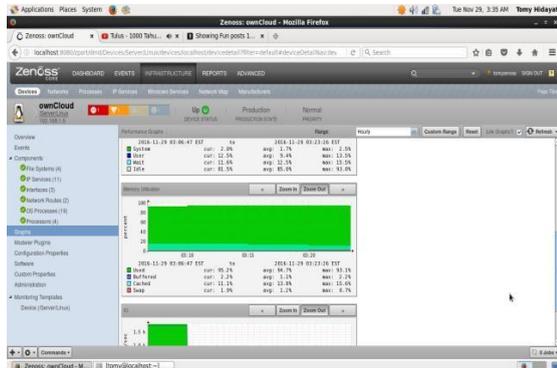
Berikut tampilan dari grafik cpu setelah *server* menjalankan streaming pada browser.



Gambar 4.69 Grafik CPU saat streaming

Gambar diatas menunjukkan bahwa grafik dari status idle yang berwarna abu-abu menurun dari menit awal hingga jam 3.23 am waktu *server*.

Kemudian dari status user yang berwarna biru mengalami peningkatan sedikit dikarenakan streaming browser yang membutuhkan data cpu lebih banyak.

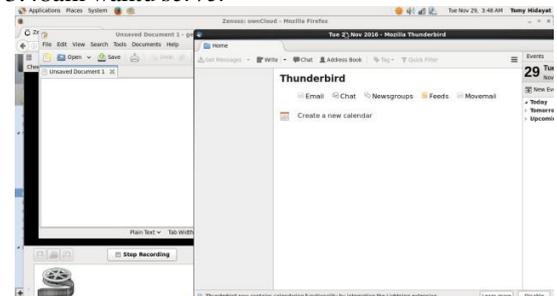


Gambar 4.70 Grafik Memori Saat Streaming

Gambar diatas menunjukkan bahwa grafik dari status used di memori sedikit meningkat dari jam 3.06 am hingga jam 3.23 am waktu server. Kemudian grafik status cache dari grafik memory sedikit menurun dari jam 3.06 am hingga 3.23 am waktu server

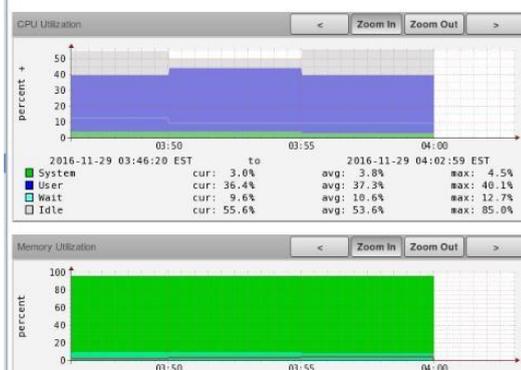
**Kasus 2**

Berikut kasus 2, jika server membuka aplikasi lebih dari satu. Waktu pada gambar menunjukkan 3.48am waktu server



Gambar 4.71 Kasus 2 Membuka Banyak Aplikasi

Berikut beberapa hasil grafik dari cpu dan memori pada kasus 2.

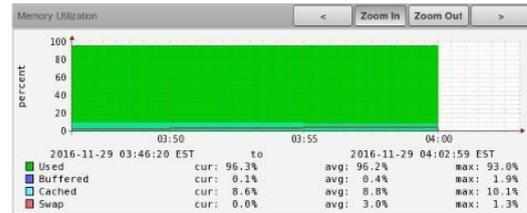


Gambar 4.72 Grafik Keterangan CPU Kasus 2

Gambar diatas menunjukkan waktu grafik yang diambil sekitar jam 3.46. am hingga 4.02. am waktu server.

Terlihat perubahan status idle pada jamm 03.50 am hingga 03.55 am menurun, kemudian naik kembali pada menit berikutnya.

Kemudian pada status sistem terjadi penurunan dari jam 03.55 am hingga menit berikutnya. Status wait menurun beberapa persen pada jam 03.50 am hingga menit berikutnya. Status user berubah pada jam 03.50 hingga 03.55 kemudian menurun pada menit berikutnya.



Gambar 4.73 Grafik Status Memori Pada Kasus 2

Gambar tersebut menunjukkan grafik memori pada jam 03.46 am hingga 04.00 am pada waktu server.

Pada status **used** tidak ada perubahan grafik memori.

Pada status **cached** sedikit menurun pada jam 03.55 am.

Pada status **swap** terjadi peningkatan bertahap pada jam sebelum 03.46 hingga 03.50, kemudian meningkat pada 5 menit berikutnya, dan meningkat lagi pada 5 menit berikutnya.

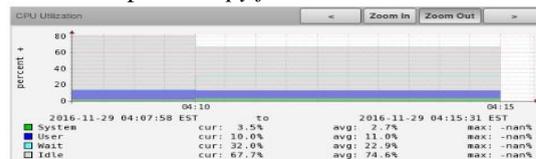
**Kasus 3**

Pada kasus ini akan mencoba bagaimana perubahan grafik pada saat copy file.

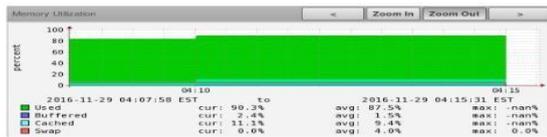


Gambar 4.74 Proses Copy File

Gambar diatas menunjukkan jam 4.20 am pada waktu server. Selama empat menit server melakukan proses copy file.

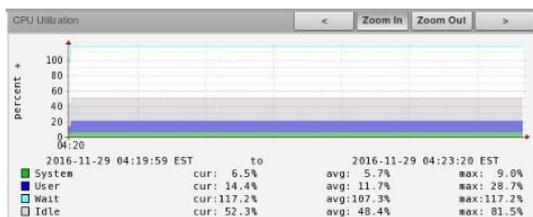


Gambar 4.75 Grafik Status Awal CPU Kasus 3



Gambar 4.76 Grafik Status Awal Memory Kasus 3

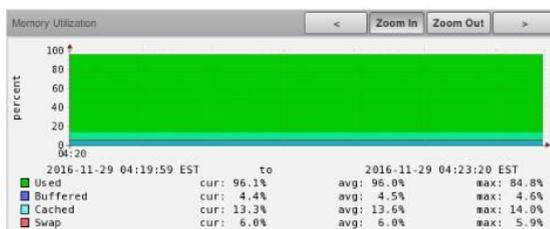
Gambar diatas menunjukkan status awal grafik sebelum melakukan proses *copy file*. Setelah melakukan copy file, perubahan grafik ditunjukkan seperti gambar dibawah.



Gambar 4.77 Grafik Status CPU Saat Kasus 3

Dari gambar diatas dapat diambil keterangan sebagai berikut disaat melakukan proses *copy file*.

Status **system** naik 3% dari status awal. Status awal cpu system sebesar 3.5%  
 Status **user** meningkat 4% dari status awal. Status awal cpu user sebesar 10.4%.  
 Status **wait** meningkat drastis hampir 60%. Status awal cpu wait sebesar 32.0%.  
 Status **idle** menurun 15%. Status awal cpu idle sebesar 67.0%.



Gambar 4.78 Grafik Status Memori Kasus 3

Dari gambar diatas dapat diambil keterangan sebagai berikut disaat melakukan proses *copy file*.

Status **used** meningkat 6%. Status awal memory used sebesar 90.0%.  
 Status **buffered** meningkat 2%. Status awal memory buffer sebesar 2.4%.  
 Status **cached** meningkat 2%. Status awal memory cached sebesar 11.3%  
 Status **swap** meningkat 6%. Status awal memory swap sebesar 0%.

## 5. Kesimpulan

### 5.1 Kesimpulan

Kesimpulan yang diperoleh dari proyek akhir ini adalah sebagai berikut :

1. Sinkronisasi file dari *private cloud* ownCloud telah berhasil yaitu user (admin atau client) dapat membuat file share antar user (antar admin, antar client atau admin dengan client) dan mengkonfigurasi fitur aplikasi desktop owncloud yang terdapat di client.
2. Sinkronisasi kontak dari *private cloud* ownCloud telah berhasil dibuat dengan share data antar kontak. Dengan adanya fitur share daftar kontak, daftar kontak user admin atau client dapat dibagikan ke user lain dengan mudah secara sinkron.
3. Sinkronisasi kalender dari *private cloud* ownCloud telah berhasil dibuat dengan membuat *share event* antar kontak dan dapat mengatur detail dari *share event* tersebut.
4. Monitoring komputer lokal dari *private cloud* dapat dilakukan dengan aplikasi zenoss terdapat grafik untuk penggunaan memori *server* dan penggunaan CPU *server*.

## 5.2 Saran

Saran dari proyek akhir untuk pengembangan selanjutnya adalah

1. Dapat ditambahkan verifikasi monitoring dengan email alert.
2. Dapat ditambahkan fitur sign up untuk penggunaan cloud.
3. Dapat ditambahkan fitur chat antar kontak cloud.

## Daftar Pustaka

Semua rujukan yang tercantum dalam daftar pustakak harus dirujuk dalam pembahasan, sehingga daftar pustaka hanya memuat pustaka yang dirujuk dalam pembahasan.

Pernyataan dalam pembahasan yang merujuk kepada pustaka diberikan keterangan perujukan dengan menggunakan nomor pustaka sesuai yang tercantum pada daftar pustaka dan ditulis dalam kurung siku, seperti [1],[2,5-7].

Daftar pustaka dituliskan dengan menggunakan huruf Times New Roman berukuran 8 pts.

- [1] George, T. (25,November 2013). *5 Reason Why You Should Create Your Own Cloud*. Dipetik Desember 23, 2016, dari Cloudwards: <https://www.cloudwards.net/5-reason-create-cloud/>
- [2] GmbH, o. (2011). Owncloud Features. Dipetik Desember 23, 2016, dari Owncloud Features: <https://owncloud.com/features/>
- [3] Sofana, I. (2012). Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS). In I. Sofana, *Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS)* (p. 13). Bandung: Informatika Bandung.
- [4] Collaboration, N. P. (n.d.). *Cloud Storage*. Retrieved Maret 21, 2016, from <http://www.baylor.edu/business/mis/nonprofit/doc.php/197132.pdf>
- [5] OwnCloud. (n.d.). ownCloud Overview. Retrieved 3 27, 2016, from ownCloud: <https://owncloud.com/owncloud-overview/>

- [6] Sofana, I. (2012). Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS). In I. Sofana, *Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS)* (p. 13). Bandung: Informatika Bandung.
- [7] Setiawan, P.(t,thn). Pengertian Sinkronisasi Sistem Operasi Lengkap. Dipetik Desember 7, 2016 dari GuruPendidikan.com:  
<http://www.gurupendidikan.com/pengertian-sinkronisasi-sistem-operasi-lengkap/>
- [8] M, S. (2015, Oktober 27). *60 Menit Belajar Sistem Monitoring Cacti - dokumen*. Retrieved April 5, 2016, from Doc Slide: <http://dokumen.tips/documents/60-menit-belajar-sistem-monitoring-cacti.html>
- [9] Zenoss, I. (2014). *Zenoss\_Core\_Installation\_Upgrade\_01-062014-4.2-v20.pdf*. Dipetik 10 4, 2016, dari Zenoss Core:  
[https://www.zenoss.com/documentation/Core\\_Docs/Zenoss\\_Core\\_Installation\\_Upgrade\\_01-062014-4.2-v20.pdf](https://www.zenoss.com/documentation/Core_Docs/Zenoss_Core_Installation_Upgrade_01-062014-4.2-v20.pdf)