

MIRRORING CLOUD STORAGE MENGGUNAKAN OWNCLOUD UNTUK AKSES, SINKRONISASI, DAN SHARE DATA

Ahmad Sholehudin

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya, ahmadsholehudin25@gmail.com

Asmunin

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya, asmunin@yahoo.com

Abstrak

Cloud storage merupakan perkembangan teknologi penyimpanan data berbasis awan, dan ownCloud merupakan salah satu teknologi dalam bidang tersebut. Teknologi *cloud storage* memungkinkan pengguna melakukan penyimpanan tanpa memerlukan penyimpanan fisik. Namun dibalik keunggulannya *cloud storage* juga memiliki kekurangan yaitu batas akses dari pengguna menuju.

Mirroring cloud storage dilakukan dengan sinkronisasi pada level *database* dan dokumen yang dilakukan secara bergantian setiap satu jam. Proses sinkronisasi dilakukan menggunakan *backup database* kemudian dijalankan berdasarkan jadwal pada *crontab*, sedangkan pada level dokumen dilakukan dengan menggunakan *rsync* dan dijalankan berdasarkan jadwal pada *crontab*.

Berdasarkan pada tugas akhir ini *mirror server* dapat digunakan sebagai sinkronisasi, akses dan share data, namun proses share tidak dapat dilakukan di *server* tujuan sinkronisasi. Dengan waktu sinkronisasi yang disesuaikan yaitu setiap tiga menit pada level dokumen dan setiap menit pada level *database*. Terdapat kekurangan yaitu jika upload data bersamaan pada jadwal sinkronisasi maka data akan di sinkronisasi pada tiga menit selanjutnya.

Kata Kunci : *Cloud Storage, Mirror server, OwnCloud*

Abstract

Cloud storage is the development of cloud-based data storage technology, and ownCloud is one of the technologies in the field. Cloud storage technology allows the user to store without the need for physical storage. But behind the advantages of cloud storage also have shortcomings that limit the access of users reached.

Mirroring cloud storage with synchronization performed at the database level and document performed alternately every hour. Synchronizing using database backup and then run based on a schedule *crontab*, while at the document level is done by using *rsync* and executed based on the schedule *crontab*.

Based on this thesis can be used as a mirror server synchronization, access and share data, but the share can not be done on the destination server synchronization. With time synchronization is adjusted every three minutes at the document level and every minute at the database level. There is a disadvantage that if the upload of data simultaneously on the synchronization schedule, the data will be in sync on the next three minutes.

Keywords: *Cloud Storage, Mirror server, OwnCloud*

PENDAHULUAN

Teknologi informasi yang sedang berkembang pesat pada sekarang ini menuntut kita untuk tetap berada sejalan dengan perkembangannya, karena dengan kita sejalan dengan perkembangan teknologi informasi maka segala kegiatan yang kita lakukan akan sedikit demi sedikit dimudahkan oleh perkembangan teknologi tersebut. Termasuk dengan perkembangan teknologi penyimpanan data yang pada era ini sudah berkembang dengan penyimpanan data berbasis internet yang lebih dikenal sebagai *cloud storage*. Teknologi *cloud storage* sangat memudahkan kita karena tanpa harus membawa

flashdisk ataupun *compact disc* kemana mana. Salah satu perusahaan yang bergerak pada teknologi *cloud storage* adalah ownCloud, ownCloud sendiri memiliki tampilan yang sangat bersahabat bagi para pengguna.

Dibandingkan dengan menggunakan media penyimpanan fisik penggunaan *cloud storage* bisa dibilang memiliki beberapa keunggulan. Namun meskipun dengan banyaknya keunggulan yang dimiliki bukan berarti teknologi *cloud storage* tidak memiliki kekurangan didalamnya, salah satunya adalah batas kemampuan *server* dalam menampung akses pengguna dalam waktu yang bersamaan.

Karena kekurangan itulah muncul terobosan baru seperti sinkronisasi antar dua *server* yang memungkinkan setiap *server* saling bertukar data dan memberikan lebih banyak pilihan lokasi akses dari pengguna menuju *server*, yang pada saat ini lebih dikenal sebagai *mirror server*.

Tujuan dari judul *mirroring cloud storage* menggunakan owncloud untuk akses sinkronisasi, dan *share data* adalah dapat merancang dan juga *mirroring cloud storage* menggunakan dua buah *server* owncloud dan *cloud storage* wnccloud digunakan untuk akses, sinkronisasi dan *share data*. Manfaat dari *mirroring cloud storage* menggunakan owncloud untuk akses sinkronisasi, dan *share data* adalah *Mirroring servercloud storage* diharapkan mampu meringankan beban kerja dari *server* dan *cloud storage* mampu digunakan untuk membantu manajemen file seperti untuk akses, sinkronisasi, dan *share data*.

KAJIAN PUSTAKA

Cloud computing

Komputasi awan (*Cloud Computing*) adalah sebuah model atau komputasi/computing, dimana sumber daya seperti *processor/computing power*, *storage*, *network*, dan *software* menjadi *abstrak* dan diberikan sebagai layanan di jaringan internet menggunakan pola akses remote. Model *billing* dari layanan ini umumnya mirip dengan model layanan *public* (Purbo, Onno W. 2014).

Klasifikasi layanan komputasi awan

Sebuah setup infrastruktur model *Cloud Computing* biasanya dikenal sebagai '*cloud*'. Berikut adalah beberapa kategori layanan yang tersedia dari sebuah '*cloud*' seperti:

1. Software as a Service (SaaS)

SaaS ini merupakan layanan *Cloud Computing* yang paling dahulu populer. *Software as a Service* ini merupakan evolusi lebih lanjut dari konsep ASP (*Application Service Provider*). Sesuai namanya, SaaS memberikan kemudahan bagi pengguna untuk bisa memanfaatkan sumberdaya perangkat lunak dengan cara berlangganan sehingga tidak perlu mengeluarkan investasi, baik untuk *in house development* ataupun pembelian lisensi.

Dengan cara berlangganan via web, pengguna dapat langsung menggunakan berbagai fitur yang disediakan oleh penyedia layanan dengan konsep SaaS ini, pelanggan tidak memiliki kendali penuh atas aplikasi yang mereka sewa, tetap hanya mengendalikan fitur-fitur aplikasi yang telah disediakan oleh penyedia saja (Purbo, Onno W. 2014).

2. Platform as a Service (PaaS)

Seperti namanya, PaaS adalah layanan yang menyediakan modul-modul siap pakai dan dapat digunakan untuk mengembangkan sebuah aplikasi (yang tentu saja hanya bisa berjalan di atas *platform* tersebut). Seperti layanan SaaS, pengguna PaaS tidak memiliki kendali terhadap sumber daya komputasi dasar seperti memori, media penyimpanan, *processing power*, dan lain-lain. Semuanya akan diatur oleh provider layanan ini. Pionir di area ini adalah Google AppEngine yang menyediakan berbagai *tools* untuk mengembangkan aplikasi di atas *platform* Google dengan menggunakan bahasa pemrograman Python dan Django. (Purbo, Onno W. 2014)

3. Infrastructure as a Service (IaaS)

IaaS terletak satu level lebih rendah dibanding PaaS. IaaS adalah sebuah layanan yang "menyewakan" sumber daya teknologi informasi dasar, yang meliputi media penyimpanan, *processing power*, *memory*, *system* operasi, kapasitas jaringan, dan lain-lain. Dapat digunakan oleh penyewa untuk menjalankan aplikasi yang dimilikinya. Model bisnisnya mirip dengan penyedia data *center* yang menyewakan ruangan untuk *co-location*, tapi ini lebih ke level mikronya. Penyewa tidak perlu tahu dengan mesin apa dan bagaimana caranya penyedia layanan menyediakan layanan IaaS. (Purbo, Onno W. 2014)

Model Cloud Computing

Setelah kita tahu jenis layanan dari *cloud computing*, sekarang kita bahas tentang *deployment model* dari *cloud computing*. Menurut NIST, ada empat *deployment model* dari *cloud computing* ini, yaitu:

1. Public Cloud

Public cloud adalah layanan *Cloud Computing* yang disediakan untuk masyarakat umum. Pengguna bisa langsung mendaftar ataupun memakai layanan yang ada. Banyak layanan *Public Cloud* yang gratis, dan ada juga yang perlu membayar untuk bisa menikmati layanannya.

Adalah layanan *cloud computing* yang disediakan untuk memenuhi kebutuhan internal dari organisasi/perusahaan. Biasanya departemen IT akan berperan sebagai *service provider* (penyedia layanan) dan departemen lain menjadi *service consumer*. Sebagai *service provider*, tentu saja Departemen IT harus bertanggung jawab agar layanan bisa berjalan dengan baik sesuai dengan standar kualitas layanan yang telah ditentukan oleh perusahaan, baik infrastruktur, platform, maupun aplikasi yang ada. (Alex Budiyo, 2012)

2. Private Cloud

Private cloud adalah layanan *cloud computing* yang disediakan untuk memenuhi kebutuhan internal dari organisasi/perusahaan. Biasanya departemen IT akan berperan sebagai *service provider* (penyedia layanan) dan departemen lain menjadi *service consumer*. Sebagai *service provider*, tentu saja Departemen IT harus bertanggung jawab agar layanan bisa berjalan dengan baik sesuai dengan standar kualitas layanan yang telah ditentukan oleh perusahaan, baik infrastruktur, platform, maupun aplikasi yang ada. (Alex Budiyo, 2012)

3. Hybrid Cloud

Hybrid cloud adalah gabungan dari layanan *public cloud* dan *private cloud* yang diimplementasikan oleh suatu organisasi/perusahaan. Dalam *Hybrid Cloud* ini, kita bisa memilih proses bisnis mana yang dipindahkan ke *public cloud* dan proses bisnis mana yang harus tetap berjalan di *private cloud*. (Alex Budiyo, 2012)

4. Community Cloud

Community cloud adalah layanan *Cloud Computing* yang dibangun eksklusif untuk komunitas tertentu, yang *consumer*-nya berasal dari organisasi yang mempunyai perhatian yang sama atas sesuatu/beberapa hal, misalnya saja standar keamanan, aturan, *compliance*, dsb. *Community Cloud* ini bisa dimiliki, dipelihara, dan dioperasikan oleh satu atau lebih organisasi dari komunitas tersebut, pihak ketiga, ataupun kombinasi dari keduanya. (Alex Budiyo, 2012)

Cloud Storage

Cloud Storage adalah sebuah teknologi penyimpanan data digital yang memanfaatkan adanya *server* virtual sebagai media penyimpanan. Tidak seperti media penyimpanan perangkat keras pada umumnya seperti compact disc atau hard disc, teknologi *Cloud storage* tidak membutuhkan perangkat tambahan apapun. Yang anda perlukan untuk mengakses file digital anda hanyalah perangkat komputer atau gadget yang telah dilengkapi layanan internet. (Tri Adi Nugroho, 2014)

Mirror server

Mirror server atau disebut juga sinkronisasi *server* merupakan duplikat dari suatu *server* yang berbeda dengan *server* utama. Fungsi *mirror* adalah untuk mengurangi kemacetan data pada suatu situs. *Server* utama merupakan *server* yang paling akurat dan *mirror* bertugas untuk menyalin isi *server* utama tersebut. (Hermawan andri dkk, 2012)

OwnCloud

OwnCloud merupakan suatu perusahaan dengan proyeknya yaitu OwnCloud project. Slogan perusahaan ini adalah *Your Cloud, Your Data, Your Way!*. OwnCloud yang merupakan salah satu perangkat lunak berbagi berkas gratis dan bebas seperti Dropbox, menyediakan pengamanan yang baik, memiliki tata cara yang baik bagi pengguna aplikasi untuk membagi dan mengakses data yang secara lancar terintegrasi dengan perangkat teknologi informasi yang tujuannya mengamankan, melacak, dan melaporkan penggunaan data. (OwnCloud, 2016)

METODE

Analisa Sistem

Sistem *cloud storage* merupakan sistem penyimpanan terbaru dimana penyimpanan *cloud storage* yang berbasis komputasi awan memungkinkan penggunaannya untuk tidak harus membawa penyimpanan fisik kemana-mana. Karena pada sistem *cloud* ini pengguna hanya diharuskan terhubung dengan internet ataupun setidaknya berada pada satu jaringan yang sama untuk bisa mengakses *cloud* tersebut. Ini sangat memudahkan pengguna karena ketersediaannya jaringan internet pada saat ini sudah hampir merata di setiap tempat jadi untuk akses ke *cloud* tersebut bisa dilakukan di setiap tempat asalkan masih tersedia jaringan internet/intranet.

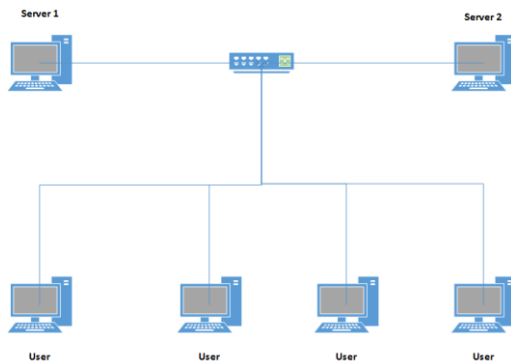
Cloud storage sendiri dibalik banyaknya keunggulan juga memiliki kekurangan didalamnya dimana salah satunya adalah batas kemampuan *server* dalam menerima akses dari pengguna yang menyebabkan kinerja dari *server* akan melambat dan mengganggu kenyamanan pengguna. Namun dengan berkembangnya teknologi keluarlah sebuah solusi untuk masalah akses tersebut. Yaitu dengan melakukan sinkronisasi antar *server* yang memungkinkan pengguna untuk memiliki tambahan pilihan akses ke *server* dan membuat beban dari *server* lebih ringan.

Desain Sistem

Sistem *mirror server* pada tugas akhir ini dilakukan pada level database dan juga pada level data. Dimana pada level database digunakan perintah `mysqldump` yang memindahkan semua database dan pada level data digunakan perintah `rsync` yang keduanya dijalankan menggunakan penjadwalan `crontab`, pada proses penjadwalan database dilakukan setiap menit dan pada proses penjadwalan data dilakukan setiap 3 menit dimana pada satu menit sebelum proses sinkronisasi dilakukan terlebih dahulu dilakukan perintah mengubah hak akses dari folder dan juga file `owncloud.log` untuk memungkinkan proses sinkronisasi dilakukan kemudian

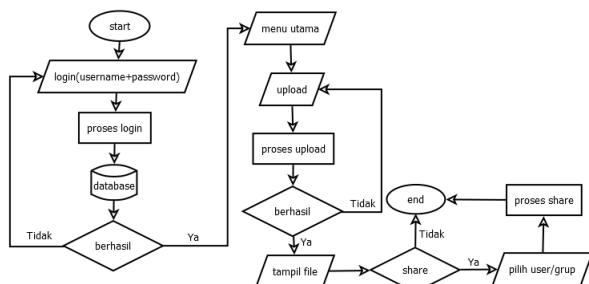
pada satu menit setelah proses sinkronisasi dilakukan proses mengubah kembali hak akses dari folder data dan juga file owncloud.log.

Berikut adalah desain sistem yang akan dibangun untuk memenuhi kebutuhan manajemen file menggunakan ownCloud yang telah dilakukan *mirror* terhadap kedua *server*nya.



Gambar 1. Desain sistem

Berikut pada gambar diatas menjelaskan bagaimana aplikasi owncloud bekerja untuk akses, *share* dan juga sinkronisasi data. Pada gambar diatas hanya digambarkan secara sederhana bagaimana langkah upload data kemudian data akan tersinkronisasi dengan folder client yang telah memasang aplikasi owncloud client dan bagaimana langkah *share* data menggunakan owncloud dengan pengguna dan grup.



Gambar 2. Flowchart penggunaan owncloud

HASIL DAN PEMBAHASAN

Pembahasan

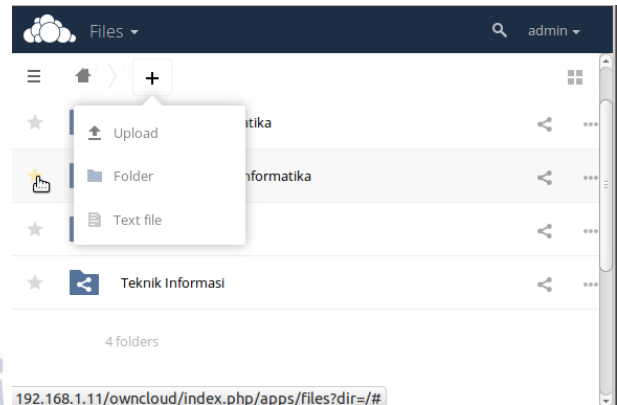
Pada pembahasan kali ini akan dijelaskan bagaimana cara kerja dari cloud *server* yang telah dibuat¹, pada bab sebelumnya.

Proses upload

Untuk proses upload data pengguna tinggal memilih tombol berbentuk lambang + pada layar atas ownCloud, setelah itu bisa dilanjutkan dengan memilih file yang akan di upload seperti pada gambar 3.

Pada proses ini saya melakukan proses upload menggunakan format .odt dan berhasil dilakukan pada

kedua buah *server*, untuk beberapa format lain yang bisa diupload pada kedua *server* terdapat pada tabel 1 dan tabel 2.

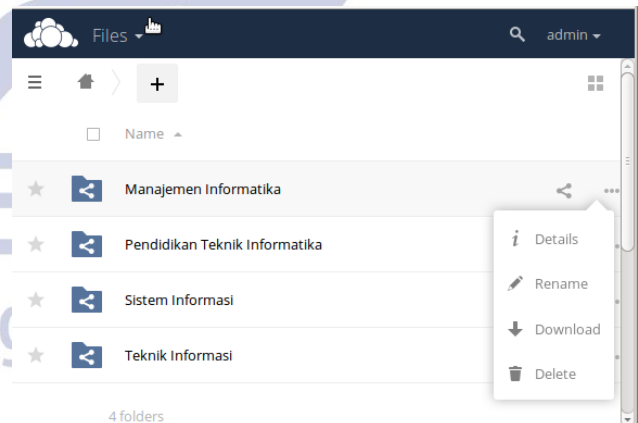


Gambar 3. Tampilan Upload Data

Proses download

Untuk proses download pengguna tinggal memilih tombol pilihan dengan bentuk 3 titik kemudian memilih download, setelah itu pengguna akan diminta untuk memilih antara membuka file atau download file, setelah pilih download maka proses download akan berlangsung, terlihat pada gambar 4.

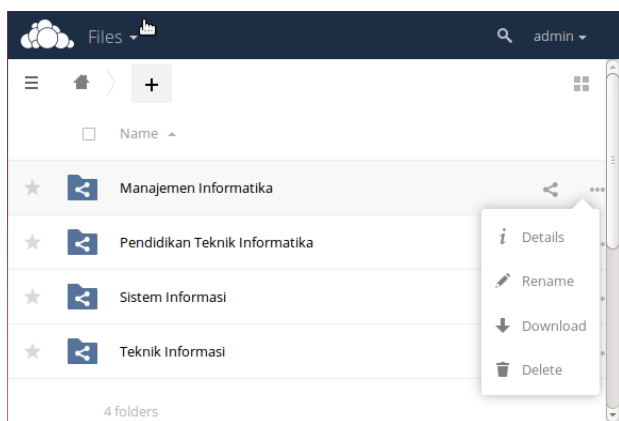
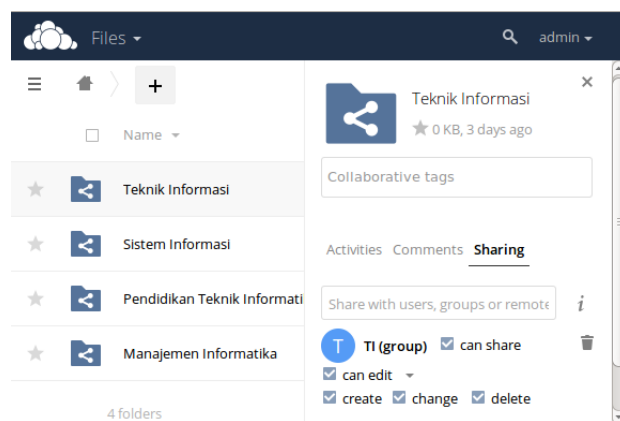
Pada proses download ini saya melakukan proses download menggunakan format .odt dan berhasil pada kedua *server*, untuk format lain yang bisa didownload pada ke dua *server* bisa dilihat pada tabel 1 dan 2.



Gambar 4. Tampilan Download Data

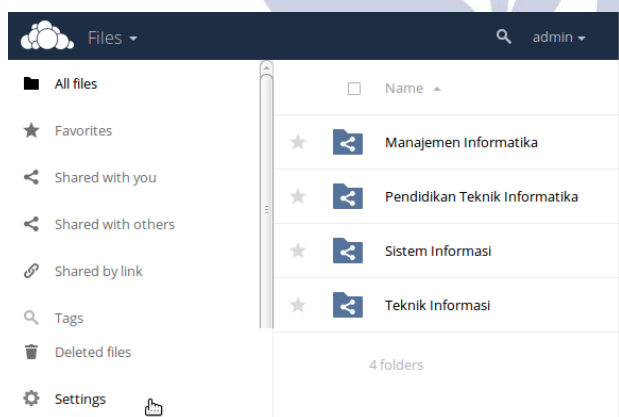
Proses hapus

Untuk proses hapus ini sama seperti proses download diatas dimana dengan menggunakan tombol yang sama namun pada pilihan terakhir memilih *delete* untuk melanjutkan prosesnya, seperti pada gambar 5

Gambar 5. Tampilan *Delete Data*Gambar 7. Tampilan *Share Data*

Proses *restore*

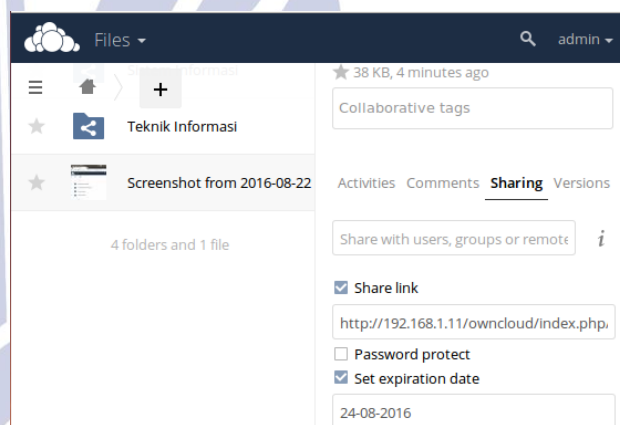
Untuk proses ini pengguna tinggal memilih pilihan pada tombol 3 garis disebelah kiri atas kemudian memilih delete file, maka setelah itu akan ditampilkan file yang pernah dihapus, kemudian pengguna tinggal memilih tombol restore untuk mengembalikan file yang telah dihapus, namun untuk *server mirroring* ini setiap *server* tidak bisa melakukan restore jika proses hapus data dilakukan dari *server* lain, maka proses restore hanya bisa dilakukan oleh *server* tempat dimana file dihapus, seperti pada gambar 6.

Gambar 6. Tampilan *Restore Data*

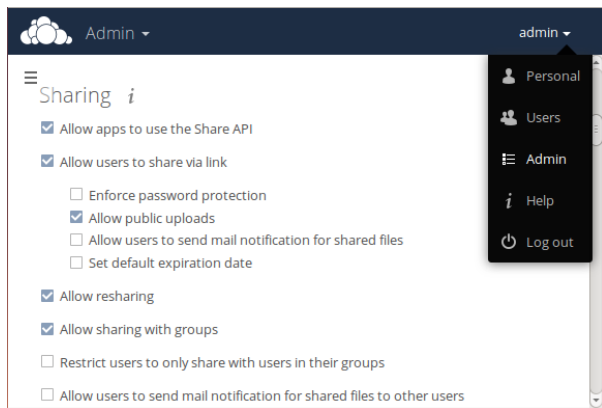
Proses *share*

Untuk proses *share* data ataupun folder pengguna hanya tinggal memilih tombol share pada setiap baris lokasi file, kemudian memberikan nama pengguna ataupun grup mana yang akan menerima *share* file tersebut, bisa juga ditambahkan pengaturan tambahan seperti apakah pengguna tersebut bisa melakukan *re-share* data, edit data, membuat data baru, melakukan perubahan versi data terbaru, dan juga menghapus data seperti pada gambar 7.

Share data juga bisa dilakukan dengan pilihan public *share* dengan menggunakan pilihan *share* link, pada *share* public ini bisa dilakukan penambahan tanggal kadaluarsa untuk link yang bisa di akses, terlihat pada gambar bahwa link hanya akan bisa diakses sebelum tanggal 24 agustus 2016 , lihat pada gambar 8

Gambar 8. Tampilan *Share Public*

Pada proses *share* data ini terdapat beberapa pengaturan tambahan seperti memberikan password, memberikan default tanggal kadaluarsa (hanya untuk public *share*), dan juga masih banyak lagi, untuk melakukan pengaturan tambahan admin hanya tinggal pilih nama akun > pilih admin > atur pada bagian sharing, terlihat pada gambar 8.



Gambar 9. Tampilan Pengaturan Sharing Data

Proses mirror server

Untuk proses yang satu ini setiap *server* telah diberikan sebuah jadwal dimana proses sinkronisasi akan terjadi pada menit ke 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57 0. Dan dilakukan secara bergantian setiap satu jam sekali. Seperti pada lima proses diatas maka setiap proses hanya akan di sinkronasikan ke *server* tujuan setiap pada menit yang telah dijadwalkan diatas.

Pemilihan waktu juga berdasarkan dengan pertimbangan sebelumnya, seperti ketika menggunakan jadwal setiap menit setiap proses dalam perintah crontab akan dijalankan dalam waktu yang bersamaan yang mengakibatkan perintah buka hak akses dari file owncloud.log pada owncloud, sinkronisasi dari folder data, dan juga perintah menutup hak akses dari file owncloud.log tidak bisa berjalan secara berurutan, dan mengakibatkan proses sinkronisasi tidak berjalan sesuai jadwal, yaitu terjadi setiap dua menit sekali tidak seperti jadwal yang telah dibuat yaitu setiap satu menit. Proses *mirror server* ini juga tidak memungkinkan untuk dilakukan penjadwalan setiap 2 menit karena masalah yang muncul juga sama yaitu proses sinkronisasi tidak berjalan sesuai dengan jadwal yang telah dibuat. Dan dari semua proses sinkronisasi di 5 nomor diatas tidak akan berjalan jika lima proses diatas dilakukan pada *server* yang berstatus sebagai tujuan.

Proses sinkronisasi

Pada proses sinkronisasi antara client dan juga server akan terjadi ketika client melakukan upload, mendapat *share* dari pengguna lain, dan juga menghapus data. Maka akan secara otomatis proses sinkronisasi terjadi. Dan untuk mematikan proses sinkronisasi client bisa keluar dari aplikasi *client* sinkronisasi.

Proses edit data real-time

Untuk proses edit data real-time ini *mirroring server* tidak digunakan karena hanya dilakukan oleh pengguna

dengan *server* yang sama, dan pada proses ini pengguna akan mendapatkan hasil edit terbaru yang dilakukan oleh pengguna lain dengan melakukan refresh pada browser yang digunakan. Namun proses edit data real-time ini hanya bisa dilakukan dengan menggunakan format file asli yang telah disediakan oleh ownCloud yaitu .txt. edit data *real-time* ini bagus digunakan untuk melakukan edit dalam waktu yang bersamaan.

Proses View data

Untuk proses view data, berdasarkan format yang dimiliki ownCloud, beberapa file tidak dapat dilihat secara langsung dan hanya bisa dilakukan upload dan download seperti word, ppt, excel, dan juga semua file video, dan juga audio, beberapa file yang bisa dilihat secara langsung pada owncloud seperti .pdf, .txt, dan juga beberapa format foto seperti .png, .jpg, dan .jpeg.

Uji coba

Pada uji coba ini akan dilakukan beberapa proses seperti upload, view dan juga download menggunakan beberapa format file yang berbeda dengan menyajikan hasil ke dalam sebuah tabel, lihat pada tabel 1 dan tabel 2.

Tabel 1. Tabel Percobaan Server1

Format File	Server1			Server2	
	Up	View	Down	View	Down
.txt	✓	✓	✓	✓	✓
.odt	✓	x	✓	x	✓
.ods	✓	x	✓	x	✓
.odp	✓	x	✓	x	✓
.pdf	✓	✓	✓	✓	✓
.png	✓	✓	✓	✓	✓
.jpg	✓	✓	✓	✓	✓
.3gp	✓	x	✓	x	✓
.mp4	✓	x	✓	x	✓

Tabel 2. Tabel Percobaan Server2

Format File	Server2			Server1	
	Up	View	Down	View	Down
.txt	✓	✓	✓	✓	✓
.odt	✓	x	✓	x	✓
.ods	✓	x	✓	x	✓
.odp	✓	x	✓	x	✓
.pdf	✓	✓	✓	✓	✓
.png	✓	✓	✓	✓	✓
.jpg	✓	✓	✓	✓	✓
.3gp	✓	x	✓	x	✓
.mp4	✓	x	✓	x	✓

PENUTUP

Simpulan

Berikut adalah kesimpulan yang di dapat dari hasil sistem yang telah dibuat. Dengan menggunakan 2 buah *server* dan dilakukan proses *mirroring* maka membuat setiap pengguna memiliki pilihan untuk ke *server* mana dia akan akses, seperti contoh yaitu ketika pengguna hanya ingin mengunduh dan melihat file tanpa melakukan proses mengunggah data bisa dilakukan melalui *server* yang menjadi tujuan dan juga memungkinkan kita memiliki banyak penyimpanan file tanpa harus melakukan proses upload berulang ulang. Setiap proses dari owncloud seperti sinkronisasi, akses, dan *share* data dapat dilakukan menggunakan *mirror server*, namun untuk *server* sebagai tujuan hanya proses *share* data tidak dapat dilakukan.

Saran

Server cloud yang dibangun dalam penyelesaian tugas akhir ini masih jauh dari sempurna, untuk itu *server* cloud ini diharapkan dapat menjadi bahan atau salah satu referensi bagi pembaca agar dapat terciptanya yang lebih baik lagi. Beberapa saran yang diperlukan antara lain:

1. Melakukan proses *mirroring* secara real-time.
2. Membangun *server* cloud secara online agar akses yang dilakukan bisa dari manapun.
3. Melakukan penambahan aplikasi agar dapat membuka file seperti .ods, .odt, .odp, dan .mp4, yang juga diperlukan untuk proses belajar mengajar.

DAFTAR PUSTAKA

- Arbie. 2003. Manajemen *Database* dengan MySQL. Andi Offset
- Bowo, Adi.2016. <http://adibowo.com/komputer-server-pengertian-jenis-dan-fungsi-server/>
- Budiyanto, Alex. 2012. Pengantar Cloud Computing. Judul Tidak Diterbitkan. Cloud Indonesia
- Hermawan, Andri dkk. 2012. Rancang Bangun *Server* Repositori Perangkat Lunak Open Source Di Universitas Widya Dharma Klaten. Judul tidak diterbitkan.Yogyakarta : Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Menggunakan ownCloud.Judul tidak diterbitkan. Surakarta : Universitas Muhammadiyah Surakarta.
- Nugroho, Tri Adi. 2014. Perancangan Private Cloud Storage
<https://owncloud.org/features/> diakses pada 1 juni 2016
- Purbo, Onno W. 2014. Membuat sendiri Cloud Computing *Server* Menggunakan Open Source. Yogyakarta: ANDI.
- Solichin, Achmad. 2010. MySQL Dari Pemula Hingga Akhir. Judul tidak diterbitkan. Jakarta : Universitas Budi Luhur.
- Waloeyo, Johan Jati. 2014. Cloud Computing. Yogyakarta: ANDI.