

Kode>Nama Rumpun Ilmu : 123/Ilmu Komputer

**LAPORAN AKHIR
PENELITIAN HIBAH BERSAING**



**RANCANG BANGUN SISTEM VALIDASI KARYA ILMIAH DENGAN
MANAJEMEN DOKUMEN CLOUD**

Agus Muliantara, S.Kom, M.Kom (0016068003)

Gusti Made Arya Sasmita, S.T., M.T (0026037305)

I Made Agus Setiawan, S.Kom, M.Kom (0024018201)

UNIVERSITAS UDAYANA

NOPEMBER 2015

HALAMAN PENGESAHAN

Judul : Rancang Bangun Sistem Validasi Karya Ilmiah dengan Manajemen Dokumen Cloud

Peneliti/Pelaksana

Nama Lengkap : AGUS MULIANTARA S.Kom.,M.Kom.
Perguruan Tinggi : Universitas Udayana
NIDN : 0016068003
Jabatan Fungsional : Lektor
Program Studi : Teknik Informatika
Nomor HP : 08113882428
Alamat surel (e-mail) : muliantara@gmail.com

Anggota (1)

Nama Lengkap : GUSTI MADE ARYA SASMITA S.T., M.T
NIDN : 0026037305
Perguruan Tinggi : Universitas Udayana

Anggota (2)

Nama Lengkap : I MADE AGUS SETIAWAN S.Kom., M.Kom.
NIDN : 0024018201
Perguruan Tinggi : Universitas Udayana
Institusi Mitra (jika ada) : -
Nama Institusi Mitra : -
Alamat : -
Penanggung Jawab : -
Tahun Pelaksanaan : Tahun ke 2 dari rencana 2 tahun
Biaya Tahun Berjalan : Rp 57.500.000,00
Biaya Keseluruhan : Rp 149.236.900,00



Mengetahui,
Dekan Fakultas MIPA

(Drs. Ida Bagus Made Suaskara, M.Si.)
NIP/NIK 196606111997021001

Denpasar, 10 - 11 - 2015
Ketua,

(AGUS MULIANTARA S.Kom.,M.Kom.)
NIP/NIK 198006162005011001



Menyetujui,
Ketua Lembaga Penelitian UNUD

(Prof. Dr. Ir. I Nyoman Gde Antara, M.Eng)
NIP/NIK 196408071992031002

DAFTAR ISI

HALAMAN PENGESAHAN USUL PENELITIAN HIBAH BERSAING	i
DAFTAR ISI.....	ii
RINGKASAN	iv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Urgensi Penelitian.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1. Teknologi Cloud	5
2.2. Cloud Document Management	9
2.3. Plagiarism Checker	9
2.4. Langkah Validasi Karya Ilmiah di Lingkungan Unud.....	14
BAB 3 TUJUAN DAN MANFAAT PENELITIAN.....	18
3.1 Tujuan Khusus	18
BAB 4 METODE PENELITIAN	19
4.1. Waktu dan Tempat.....	19
4.2. Tahapan Penelitian.....	19
4.2.1. Tahapan Awal	20
4.2.2. Tahapan Pertama (tahun I).....	20
4.2.3 Tahap Kedua (tahun II).....	20
4.3 Indikator Capaian.....	22
BAB 5 HASIL YANG DICAPAI.....	23
5.1 Analisa Kebutuhan	23
5.1.1 Kebutuhan Fungsional	23
5.1.2 Kebutuhan Non Fungsional.....	23

5.2 Flow Sistem.....	24
5.3 Desain dan Implementasi	26
5.3.1 Fitur Sistem	27
5.3.2 Implementasi	27
5.3.3 Uji coba Pada LVM + AoE	41
5.3.4 Uji Coba Pada Owncloud	42
5.3.5 Ujicba User Owncloud	43
BAB 6 KESIMPULAN DAN SARAN	45
DAFTAR PUSTAKA	46

RINGKASAN

Salah satu tugas utama dosen sesuai dengan Tridharma Perguruan Tinggi adalah untuk melakukan penelitian dan menerbitkan hasil penelitiannya dalam sebuah karya ilmiah. Dalam rangka menghindari plagiarisme, Universitas Udayana menugaskan Badan Penjaminan Mutu Universitas dan masing-masing fakultas untuk melakukan validasi terhadap karya ilmiah yang dipublikasikan oleh dosen. Pada pelaksanaannya terjadi ketidakselarasan antara BPMU dan Fakultas dalam hal pemberdayaan berkas karya. Di tingkat universitas berkas karya tersebut diunggah melalui sistem, sedangkan di tingkat fakultas masih menggunakan berkas fisik. Jika proses validasi baik pada tingkat fakultas maupun universitas dapat dilakukan melalui satu pintu, maka waktu serta biaya yang dibutuhkan dapat diminimalisasi. Akan tetapi, sistem digitalisasi karya ilmiah juga memiliki kekurangan karena dibutuhkan media penyimpanan yang besar. Oleh karena itu dibutuhkan suatu sistem informasi validasi karya ilmiah yang juga didukung dengan manajemen media penyimpanan yang efisien.

Penelitian ini merupakan penelitian lanjutan dari periode 2014, yang dilaksanakan di laboratorium pemrograman Jurusan Ilmu Komputer Universitas Udayana dengan mengambil studi kasus proses validasi karya ilmiah yang dikelola dan dikoordinasikan oleh BPMU dan Lembaga Penelitian Universitas Udayana. Pada tahap ini akan dieksplorasi lebih lanjut teknologi *Cloud Computing* dan *Cloud Storage* dengan melakukan eksperimen serta penambahan fitur *plagiarism checker* untuk memberikan penilaian terhadap kesamaan artikel ilmiah yang diupload ke sistem. Adapun langkah-langkahnya adalah: Analisis Hasil Ujicoba tahap pertama, Penyesuaian desain dan arsitektur sistem, Penyesuaian implementasi kode dan sistem, Ujicoba dan Evaluasi Sistem.

Luaran yang ingin dicapai dalam penelitian ini adalah: Sistem Aplikasi Karya Ilmiah, Diseminasi Seminar Internasional, Publikasi di Jurnal Nasional Terakreditasi atau di seminar internasional

Keyword: sistem validasi karya ilmiah, Manajemen Dokumen Cloud, plagiarsim checker

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Sebagaimana diamanatkan dalam UU Nomor 14 Tahun 2005 tentang Guru dan Dosen, dosen dinyatakan sebagai pendidik profesional dan ilmuwan dengan tugas utama mentransformasikan, mengembangkan, dan menyebarluaskan ilmu pengetahuan, teknologi, dan seni melalui pendidikan, penelitian, dan pengabdian kepada masyarakat (Bab 1 Pasal 1 ayat 2). Sementara itu, profesional dinyatakan sebagai pekerjaan atau kegiatan yang dilakukan oleh seseorang dan menjadi sumber penghasilan kehidupan yang memerlukan keahlian, kemahiran, atau kecakapan yang memenuhi standar mutu atau norma tertentu serta memerlukan pendidikan profesi.

Kompetensi tenaga pendidik, khususnya dosen, diartikan sebagai seperangkat pengetahuan, keterampilan dan perilaku yang harus dimiliki, dihayati, dikuasai dan diwujudkan oleh dosen dalam melaksanakan tugas profesionalnya. Kompetensi tersebut meliputi kompetensi pedagogik, kompetensi kepribadian, kompetensi sosial dan kompetensi profesional.

Tugas utama dosen adalah melaksanakan tridharma perguruan tinggi dengan beban kerja paling sedikit sepadan dengan 12 (dua belas) sks dan paling banyak 16 (enam belas) sks pada setiap semester sesuai dengan kualifikasi akademik. Sedangkan profesor atau guru besar adalah dosen dengan jabatan akademik tertinggi pada satuan pendidikan tinggi dan mempunyai tugas khusus menulis buku dan karya ilmiah serta menyebarkan luaskan gagasannya untuk mencerahkan masyarakat. (pedoman beban kerja dosen, 2010)

Kasus plagiarisme (penjiplakan karangan) pada tahun 2010 oleh seorang guru besar merupakan tamparan bagi dunia pendidikan Indonesia sehingga harus menjadi pelajaran mengingat posisi guru besar merupakan panutan. Untuk mencegah terulangnya kasus plagiarisme. Salah satu cara yakni setiap rektor dan guru besar diimbau memberi wawasan kepada calon guru besar agar bertindak terpuji serta memiliki kesadaran individu bahwa posisi guru besar merupakan posisi terhormat baik di kalangan akademisi maupun di masyarakat. Sedangkan dari segi sistem pencapaiannya, semestinya diiringi perbaikan dengan proses seleksi yang ketat.

Universitas udayana sebagai perguruan tinggi Negeri terbesar di Bali melalui Peraturan Rektor Unud No. 2 tahun 2012 tentang validasi karya ilmiah, berusaha mencegah terjadinya plagiarism di lingkungan Universitas Udayana khususnya dan di lingkungan dosen pada umumnya. Peraturan itu menegaskan bahwa semua karya Ilmiah dosen yang akan dipergunakan untuk kebutuhan akademik (kenaikan pangkat dan jabatan akademik) harus divalidasi secara berjenjang. Proses validasi dilaksanakan mulai dari Fakultas/PPS sampai pada Universitas. Proses itu dilakukan oleh Tim Validasi Fakultas/PPS serta Tim Validasi Unud. Langkah ini diharapkan dapat menumbuhkan etika dan budaya ilmiah ke arah yang lebih baik, serta dapat mencegah terjadinya ketidak jujuran ilmiah di lingkungan Unud.

Di sisi lain, kebijakan DIKTI juga dengan tegas menyebutkan bahwa apabila terbukti ditemukan kecurangan ilmiah termasuk plagiat pada karya Ilmiah dosen yang dipergunakan untuk kenaikan pangkat, maka sanksi yang diberikan tidak hanya kepada si pelaku, tetapi juga kepada institusinya di mana seluruh kenaikan pangkat dan jabatan akademik yang terjadi pada tahun yang sama akan dianulir.

Berdasarkan hal tersebut, universitas Udayana melalui BPMU (Badan penjamin Mutu Unieversitas Udayana) menerbitkan suatu panduan atau Manual prosedur untuk melakukan proses validasi karya ilmiah tersebut. Manual Prosedur: SOP Validasi Karya Ilmiah Universitas Udayana ini mencakup langkah-langkah serta hal-hal yang perlu divalidasi pada karya ilmiah dosen di lingkungan Universitas Udayana serta tugas dan fungsi Tim Validasi Karya Ilmiah Universitas Udayana.

Dengan mengacu pada :

1. Peraturan Menteri Pendidikan Nasional Republik Indonesia No. 17 tahun 2010 tentang pencegahan dan penanggulangan plagiat di perguruan tinggi
2. Surat Edaran Dirjen Dikti no 2050/E/T/2011 tentang kebijakan unggah karya ilmiah.
3. Surat Edaran Dirjen Dikti No. 190 D/T/2011 tentang validasi karya ilmiah.
4. Peraturan Rektor Unud No 2 tahun 2012 tentang validasi karya ilmiah dosen di lingkungan Universitas Udayana
5. SK Rektor Unud No. 59/UN.14/HK/2012 tentang personalia tim validasi karya ilmiah Universitas Udayana.

Dengan adanya manual prosedur ini, diharapkan setiap dosen mengetahui dan mematuhi langkah-langkah dalam melakukan validasi publikasi karya ilmiahnya (BPMU,2012).

Besama-sama dengan manual prosedur yang telah ditetapkan oleh BPMU Universitas Udayana dan telah berakhirnya penelitian tahun pertama (2014) mengenai sistem validasi karya ilmiah di Universitas Udayana secara online, maka akan dilanjutkan pengembangan sistem validasi yang lebih mengarah pada sistem cloud dan pengujian plagiarisme secara mandiri.

Untuk itu penelitian tahun ke-2 (2015) yang akan lebih dikembangkan pada sistem cloud dan pengujian plagiarisme memiliki tujuan adalah :

1. Mampu memberi kemudahan akses dan fleksibilitas pada file yang akan divalidasi. Dimana file akan disimpan pada *cloud* sehingga akan dapat diakses dari mana saja.
2. Mampu melakukan sistem validasi berbasis reviewer. Dalam hal ini dengan cara memanfaatkan aplikasi native smartphone dalam melakukan validasi.
3. Mampu melakukan proses pengecekan plagiarisme secara otomatis.

1.2. Urgensi Penelitian

Kasus plagiarisme (penjiplakan karangan) oleh seorang guru besar merupakan tamparan bagi dunia pendidikan Indonesia sehingga harus menjadi pelajaran mengingat posisi guru besar merupakan panutan. Untuk mencegah terulangnya kasus plagiarisme diperlukan sebuah sistem validasi yang mampu memberikan hasil yang transparan, cepat, dan tentunya mencegah terjadinya plagiarisme. Apalagi dengan semakin Tingginya jumlah karya ilmiah yang dihasilkan oleh para dosen juga secara otomatis akan meningkatkan persentasi kemungkinan terjadinya plagiarism.

Karena tingginya permintaan akan proses validasi karya ilmiah yang dilakukan oleh para dosen, maka dalam penelitian ini akan dibangun model SOP validasi karya ilmiah yang optimal, sehingga proses validasi-review dapat terlaksanan secara transparan.

Disamping itu pula, dengan dilakukannya penelitian ini maka sistem validasi-review karya ilmiah dapat dilakukan lebih cepat karena seorang reviewer ataupun seorang validator tidak harus ada di ruang kerja untuk melakukan tugasnya sebagai reviewer ataupun validator. Cukup dengan menggunakan koneksi internet, sistem dapat diakses dari mana saja.

Sehubungan dengan terkumpulnya karya ilmiah dalam media penyimpana server yang dapat dianalogikan sebagai perpustakaan digital maka penelitian ini nantinya akan mampu mendokumentasikan karya ilmiah yang sudah pernah divalidasi dan juga dapat digunakan sebagai bukti klaim atas hasil suatu peneliti apabila disuatu saat terdapat klaim lain atas karya ilmiah tersebut. Disamping itu pula, dengan adanya media *cloud storage* ini, maka

keterbatasan media penyimpanan dalam suatu *server* akan dapat diatasi. Penelitian ini akan memudahkan penambahan maupun *backup* atas data karya ilmiah.

Manfaat lain dari karya ilmiah ini adalah data yang dikumpulkan dapat digunakan sebagai data set pada ranah penelitian data mining.

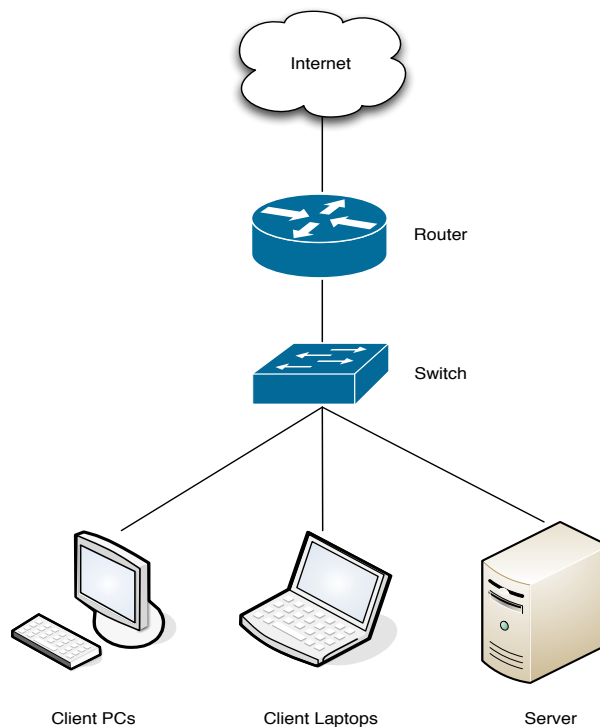
BAB 2

TINJAUAN PUSTAKA

2.1. Teknologi Cloud

2.1.1. Pengertian Cloud

Komputasi awan (*cloud computing*) saat ini mendapatkan banyak perhatian, baik publikasi maupun antar pengguna, mulai dari individu di rumah sampai ke pemerintahan. Walaupun demikian, *cloud computing* tidak selalu didefinisikan dengan jelas. *Cloud computing* merupakan layanan berbasis langganan yang menyediakan akses ke ruang penyimpanan dan sumber daya komputer yang terhubung dengan jaringan. Salah satu cara untuk memahami *cloud computing* adalah dengan memikirkan pengalaman menggunakan email. Penyedia layanan email seperti Yahoo, Gmail, Hotmail dan lain sebagainya, mengurus masalah yang berhubungan dengan perangkat lunak dan perangkat keras yang dibutuhkan untuk mendukung akun email pribadi penggunanya. Ketika pengguna ingin mengakses email maka ia hanya perlu membuka peramban di komputernya, mengetikkan alamat provider emailnya dan login. Bagian terpenting dari proses tersebut adalah tersedianya akses internet. Email pengguna tersebut tidak disimpan secara fisik pada komputer pribadinya, ia dapat mengaksesnya dengan koneksi internet dimana saja.

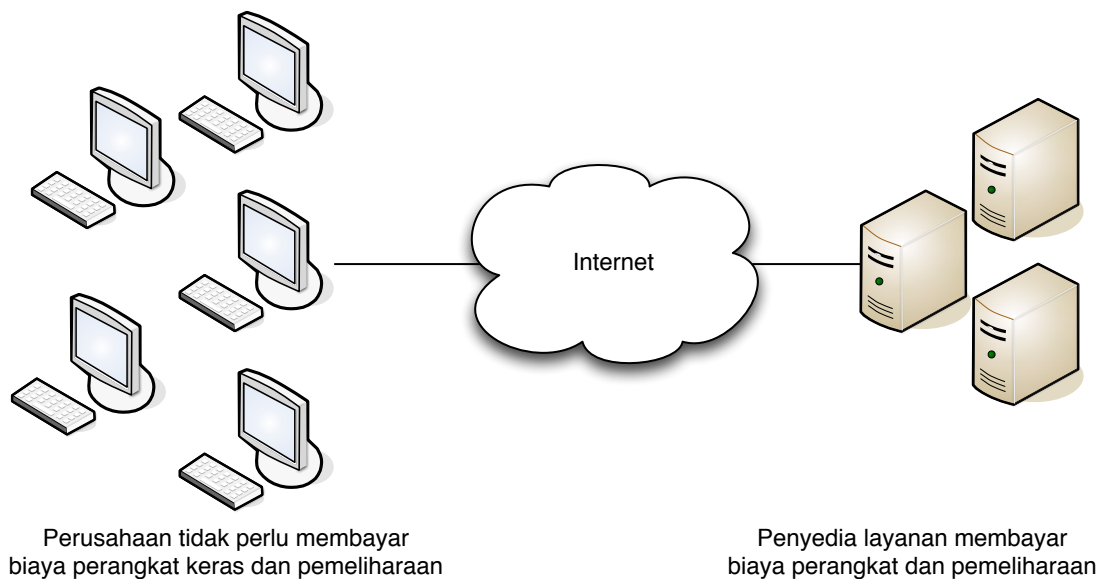


Gambar 2.1. Cloud Computing

Ketika pengguna tersebut melakukan perjalanan, sedang bekerja maupun pergi untuk membeli kopi, ia tetap dapat memeriksa emailnya selama terhubung ke internet. Email pengguna tersebut berbeda dengan aplikasi seperti aplikasi pemrosesan kata yang terpasang pada komputernya. Ketika pengguna tersebut membuat suatu dokumen menggunakan aplikasi “word” maka dokumen tersebut akan disimpan pada perangkat yang digunakan untuk membuatnya kecuali dipindahkan secara fisik. Cara kerja provider email tersebut hampir sama dengan *cloud computing*, yang berbeda hanyalah pada *cloud computing* informasi yang ingin diakses dapat dipilih.

2.1.2. Cara Kerja Cloud

Cloud memungkinkan informasi dapat diakses kapan saja dan dari mana pun. Susunan komputer tradisional mengharuskan penggunanya untuk berada pada lokasi yang sama dengan media penyimpanannya, akan tetapi *cloud* menghilangkan keharusan tersebut. Penyedia layanan *cloud* menyediakan perangkat lunak dan keras yang dibutuhkan untuk menjalankan aplikasi rumahan maupun bisnis.



Gambar 2.2. Keuntungan Menggunakan Jasa *Cloud computing*

Hal ini tentu saja sangat berguna untuk bisnis yang tidak mampu untuk mengeluarkan jumlah uang yang sama untuk perangkat keras dan media penyimpanan seperti perusahaan yang lebih besar. Perusahaan-perusahaan kecil dapat menyimpan informasinya di *cloud* sehingga menghilangkan biaya pembelian dan penyediaan perangkat memori. Sebagai tambahan, biaya dikenakan hanya untuk membayar media penyimpanan yang digunakan,

maka perusahaan dapat membeli tambahan atau mengurangi ruang ketika bisnisnya berkembang ataupun ketika keperluan untuk ruang yang lebih sedikit muncul.

Salah satu kebutuhan untuk mengakses *cloud* adalah koneksi internet. Ini berarti untuk dapat mencari suatu dokumen spesifik yang tersimpan di *cloud*, pengguna sebelumnya harus terkoneksi ke internet melalui wireless atau internet kabel maupun koneksi *mobile broadband*. Keuntungannya adalah pengguna dapat mengakses dokumennya dari mana saja menggunakan perangkat apapun yang terhubung ke internet. Perangkat-perangkat ini dapat berupa desktop, laptop, tablet ataupun smartphone. Hal ini juga dapat membantu bisnis pengguna untuk berjalan lebih mulus karena siapapun yang dapat terhubung ke internet dan mengakses akun *cloud* pengguna tersebut dapat bekerja pada dokumen, mengakses aplikasi dan menyimpan data. Sebagai ilustrasi, bayangkan seorang karyawan di suatu organisasi mengunduh dokumen pdf melalui *smartphone*-nya, mengubah dokumen tersebut tanpa harus pergi ke kantor untuk mencetaknya. Inilah kebebasan yang ditawarkan oleh *cloud computing* bagi individu ataupun organisasi.

2.1.3. Tipe-Tipe Cloud

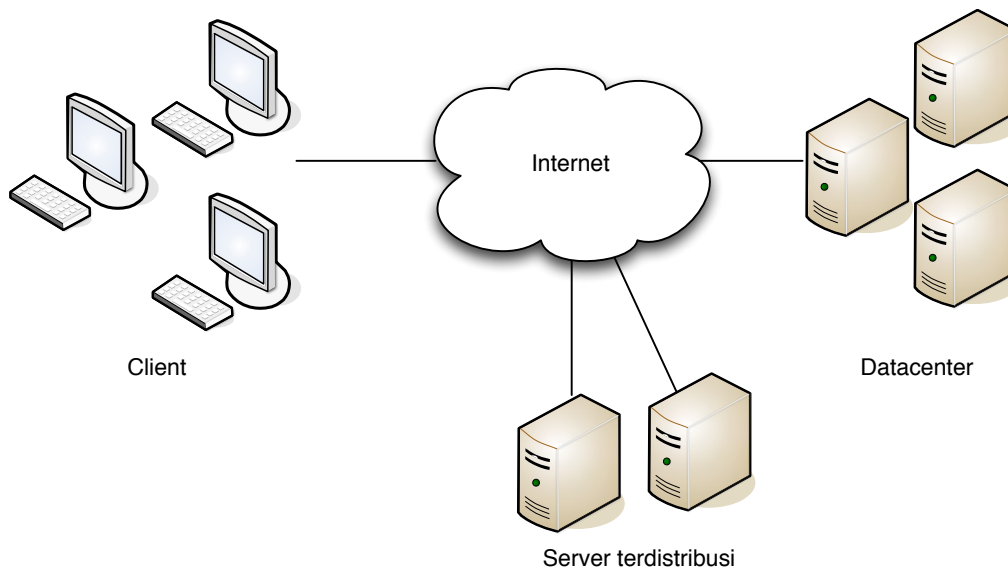
Ada beberapa tipe cloud yang berbeda yang dapat dipilih sesuai dengan kebutuhan:

- *Public cloud*, tipe ini dapat diakses oleh siapa saja yang telah mendaftar pada penyedia layanan menggunakan koneksi internet dan diberikan akses ke ruang penyimpanan di cloud.
- *Private cloud*, tipe ini disediakan hanya bagi grup atau organisasi tertentu dimana akses hanya diberikan kepada grup tersebut.
- *Community cloud*, tipe ini menggunakan ruang penyimpanan di cloud secara berbagi-pakai antara dua grup yang memiliki kebutuhan yang sama.

Hybrid cloud, tipe ini pada dasarnya merupakan kombinasi dari paling tidak dua tipe cloud yang telah disebutkan sebelumnya.

2.1.4. Komponen Cloud Computing

Secara topologi, solusi komputasi awan dibangun oleh beberapa elemen yaitu pengguna, *datacenter*, dan server terdistribusi.



Gambar 2.3. Elemen Pembangun Komputasi Awan

- **Client**

Client pada arsitektur komputasi awan adalah sama dengan *client* pada LAN (*local area network*). Awalnya *client* adalah computer yang ada di meja pengguna namun seiring dengan kemajuan teknologi, *client* disini dapat berupa laptop, komputer tablet, *mobile phone*, atau PDA. Intinya, *client* merupakan perangkat yang digunakan oleh pengguna untuk mengelola informasinya di awan. *Client* dapat dibagi menjadi tiga kategori:

1. *Mobile*, perangkat bergerak yang meliputi PDA atau ponsel pintar, seperti iPhone, Blackberry dan lainnya.
2. *Thin*, merupakan komputer yang tidak memiliki media penyimpanan internal, pekerjaan semua dilakukan di server dan hanya menampilkan informasi yang dihasilkan.
3. *Thick*, merupakan komputer biasa yang menggunakan peramban seperti Firefox atau Internet Explorer untuk mengakses awan.

- **Datacenter**

Datacenter merupakan kumpulan server dimana aplikasi yang diinginkan disimpan. *Datacenter* ini mungkin terletak di ruang bawah tanah yang penuh dengan server atau ada di belahan dunia lain yang tidak diketahui dan hanya bias diakses melalui internet.

- **Server Terdistribusi**

Server-server yang digunakan tidak harus diletakkan pada ruang yang sama. Pada umumnya server-server tersebut berada pada tempat yang secara geografis berjauhan. Tapi, di hadapan

para pengguna server-server ini seolah-olah berdampingan dan bekerja sama untuk memberikan layanan.

2.2. Cloud Document Management

Definisi dari layanan *cloud document management* terus berevolusi dan jasa layanan yang disediakan secara nyata oleh provider berbasis *cloud* meliputi serangkaian layanan penyimpanan mulai dari yang dasar sampai dengan fitur yang paling canggih. Sifat dari layanan ini juga terbuka untuk interpretasi dari vendor yang lebih menekankan kemampuan *cloud* sebagai layanannya. Sebagai contoh: kemampuan *cloud* dapat dikonsepsikan sebagai penyimpanan untuk gambar digital dari dokumen yang dipindai dan disimpan pada penyimpanan virtual yang ditawarkan oleh pihak ketiga. Penawaran lain menambahkan layanan untuk pembuatan dokumen office digital menggunakan *desktop tools*, gambar teknik dan mengelola serta berbagi pakai obyek multimedia. Unit bisnis atau departemen mungkin telah menggunakan layanan yang disediakan oleh pihak ketiga untuk aplikasi khusus yang melibatkan berbagi-pakai dokumen atau perpustakaan media untuk berbagi-pakai gambar dan grafik.

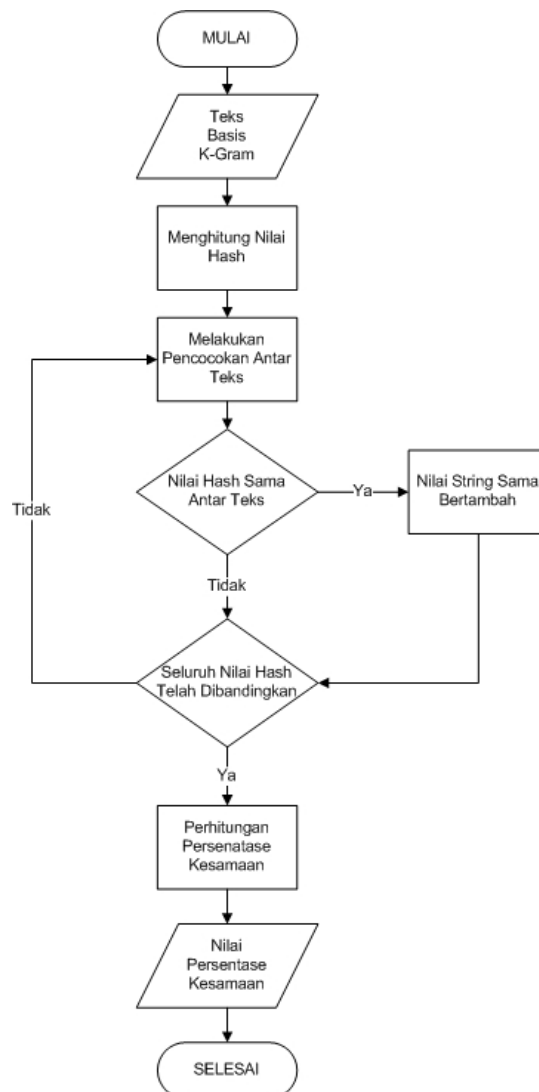
Paradigma dari komputasi awan (*cloud computing*) untuk menyimpan dan mengelola dokumen tidak dengan sendirinya merupakan obat mujarab untuk memecahkan permasalahan yang dihadapi oleh organisasi dengan sistem manajemen dokumen yang dikembangkan sendiri oleh organisasi tersebut. Inti dari tantangan yang dihadapi oleh *Chief Information Officers*, ahli teknologi dan para profesional manajemen informasi adalah untuk merencanakan, mengimplementasikan dan memelihara kemampuan manajemen dokumen yang efektif yang dapat memenuhi kebutuhan bisnis saat ini dan dapat dipertahankan.

2.3. Plagiarism Checker

Disamping penggunaan cloud storage, sistem validasi ini nantinya juga akan memiliki fitur pengujian plagiarisme atau yang disebut sebagai Sistem pendeteksi kesamaan artikel ilmiah. Sistem pendeteksi kesamaan artikel ilmiah yang dibuat ini, menggunakan algoritma yang menerapkan metode fingerprint dengan tujuan agar proses yang dilakukan berlangsung dengan cepat. Salah satu algoritma yang menggunakan teknik dokumen fingerprint adalah algoritma Rabin-Karp. Algoritma Rabin-Karp adalah suatu algoritma pencarian string yang menggunakan hashing untuk menemukan sebuah substring dalam sebuah teks (Firdaus, 2008). Algoritma Rabin-Karp tergolong algoritma multiple pattern search yang sangat efisien dalam mencari string dengan pola yang banyak, mengingat bahwa dokumen artikel ilmiah

memiliki n -string yang panjang dan pola yang beragam. Selain itu, dengan teknik dokumen fingerprint yang digunakan maka perbandingan dokumen juga berlangsung lebih cepat. Algoritma Rabin-Karp yang telah ada, akan diimplementasikan dalam sistem validasi karya ilmiah ini. Sistem pendeteksi kesamaan tersebut diharapkan mengurangi kesalahan dalam penentuan kesamaan dokumen jika dilakukan oleh manusia.

Algoritma Rabin-Karp diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* di dalam *string* teks (Gipp, 2011). Algoritma Rabin-Karp (Gambar 2.1) digunakan sebagai algoritma deteksi kesamaan pada dokumen tugas akhir. Algoritma ini memiliki beberapa keunggulan dalam penerapannya, salah satunya adalah algoritma ini sangat cocok digunakan untuk *string* yang panjang.



Gambar 2.4 Diagram Alir Algoritma Rabin-Karp

Karakteristik algoritma Rabin-Karp, yaitu

- 1) Menggunakan sebuah fungsi *hashing*
- 2) Tahap preprocessing menggunakan kompleksitas waktu $O(m)$
- 3) Untuk tahap pencarian kompleksitasnya : $O(mn)$
- 4) Waktu yang diperlukan $O(n+m)$

Dalam algoritma Rabin-Karp, ada beberapa tahap yang harus dilalui dalam implementasi algoritma tersebut.

1) Preprocessing

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang mengalami pengolahan lebih lanjut. Pembersihan yang dilakukan adalah pembersihan suatu teks dari karakter-karakter yang bersifat sebagai *stoplist*, dimana karakter tersebut banyak dan pasti muncul dalam setiap teks.

2) Rolling Hash

Fungsi yang digunakan untuk menghasilkan nilai *hash* dari rangkaian *gram* dalam algoritma Rabin-Karp adalah dengan menggunakan fungsi *rolling hash*.

Fungsi *hash* $H_{(c_1...c_k)}$ didefinisikan sebagai berikut,

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b^k + c_k$$

Keterangan :

c : nilai ASCII karakter

b : basis (bilangan prima)

k : banyak karakter

Kode Standar Amerika untuk Pertukaran Informasi atau ASCII (American Standard Code for Information Interchange) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal. Contoh bilangan ASCII dari suatu karakter adalah karakter A dengan nilai 65 atau karakter © dengan nilai 169. Dalam matematika, bilangan prima adalah suatu bilangan

asli yang lebih besar dari angka 1, faktor pembaginya adalah 1 dan bilangan itu sendiri. Angka 2 dan 3 adalah bilangan prima. Angka empat bukan bilangan prima. Sepuluh bilangan prima pertama adalah 2, 3, 5, 7, 11, 13, 17, 19, 23 dan 29.

N-grams adalah rangkaian *terms* dengan panjang N (Surahman, 2013). Kebanyakan yang digunakan sebagai *terms* adalah kata. *N-grams* merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode *n-grams* ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah n dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.. Berikut ini adalah contoh *n-grams* dengan n=6:

[1] Teks : proposal tugas akhir ilmu komputer

[2] Hasil : {propos}{roposa}{oposal}{posal}{osal t}{sal tu}{al tug}
{l tugas}{ tugas}{tugas}{ugas a}{gas ak}{as akh}{s akhi}
{akhir}{akhir}{khir i}{hir il}{ir ilm}{r ilmu}{ilmu}{ilmu k}
{lmu ko}{mu kom}{u komp}{kompu}{komput}{ompute}{mputer}

Contoh Hashing

Metode *hashing* digunakan untuk mempercepat pencarian atau pencocokan suatu *string*. Apabila tidak di-*hash*, pencarian dilakukan karakter per karakter pada nama-namayang panjangnya bervariasi dan ada 26 kemungkinan pada setiap karakter. Namun pencarian menjadi lebih mangkus setelah di-*hash* karena hanya membandingkan beberapa digit angka dengan cuma 10 kemungkinan setiap angka. Nilai *hash* pada umumnya digambarkan sebagai *fingerprint* yaitu suatu *string* pendek yang terdiri atas huruf dan angka yang terlihat acak (data biner yang ditulis dalam heksadesimal). Contoh sederhana *hashing* adalah (Firdaus, 2008):

Firdaus, Hari

Munir, Rinaldi

Rabin, Michael

Karp, Richard

menjadi

7864 : *Firdaus, Hari*

9802 : *Munir, Rinaldi*

1990 : *Rabin, Michae*

3) Pencocokan

Pada *processing*, setiap m deret (kontigu) karakter pada *field* pencarian dicari *hash key* dengan cara yang sama dengan pencarian *hash key* untuk *pattern*. Perlu diingat dengan menggunakan aturan horner, pencarian untuk m deret karakter dapat dengan mudah dihitung dari m deret karakter predesesornya. Sedangkan untuk proses pencocokannya, dipakai sebuah teorema yaitu

Sebuah stringA identik dengan stringB, jika (syarat perlu) stringA memiliki hash key yang sama dengan hash key yang dimiliki oleh stringB

4) Pengukuran Nilai Kemiripan

Mengukur *similarity* (kemiripan) dan jarak antara dua *entitas* informasi adalah syarat inti pada semua kasus penemuan informasi, seperti pada *Information Retrieval* dan *Data Mining* yang kemudian dikembangkan dalam bentuk perangkat lunak, salah satunya adalah sistem deteksi kesamaan (Salmusih, 2013). Penggunaan ukuran *similarity* yang tepat tidak hanya meningkatkan kualitas pilihan informasi tetapi juga membantu mengurangi waktu dan biaya proses sehingga mengperangkat lunakkan *Dice's Similarity Coefficient* dalam penghitungan nilai *similarity* yang menggunakan pendekatan *kgram*.

$$S = \frac{K \times C}{(A + B)}$$

Dimana S adalah nilai *similarity*, A dan B adalah jumlah dari kumpulan *n-grams* dalam teks 1 dan teks 2. C adalah jumlah dari *n-grams* yang sama dari teks yang dibandingkan. Berikut adalah contoh dari penggunaan rumus tersebut. Terdapat dua buah dokumen teks (dok A dan dok B) dengan nilai *n-gram* masing-masing dokumen secara berturut-turut adalah 2608 dan 3040, sedangkan nilai *n-gram* yang sama adalah sebesar 1203. Maka hasil nilai dari kemiripan kedua dokumen tersebut adalah

$$\begin{aligned} S &= \frac{K \times C}{(A + B)} = \frac{2 \times 1203}{(2608 + 3040)} \\ &= \frac{2406}{5648} = 0,4259 \times 100\% = 42,59\% \end{aligned}$$

2.4. Langkah Validasi Karya Ilmiah di Lingkungan Unud

1. Syarat agar karya Ilmiah dosen dapat divalidasi oleh Tim Validasi Karya Ilmiah Unud;

- a. Sudah divalidasi oleh Tim Validasi Fakultas dengan bukti keterangan validasi dari Tim Validasi Fakultas yang diketahui Dekan dilengkapi rekomendasi Ketua Program Studi.
- b. Karya Ilmiah yang diusulkan sudah didaftarkan di LPPM Unud (dibuktikan dengan surat keterangan Ketua LPPM Unud).

2. Kriteria validitas karya ilmiah

Karya ilmiah disebut valid apabila memenuhi aspek-aspek tersebut di bawah ini.

- a. Aspek kelayakan/substansi dari hasil telaah Tim Validasi Karya Ilmiah Fakultas dan Tim Validasi Karya Ilmiah Unud. Aspek kelayakan/substansi secara umum meliputi beberapa kriteria yaitu: Cakupan keilmuan karya ilmiah, Aspirasi wawasan karya ilmiah, Kemutakhiran dan *ke-pioneer-an*, Pemaparan, analisis dan sintesis karya ilmiah, Sumbangan karya ilmiah pada kemajuan ipteks, Dampak ilmiah karya Ilmiah
- b. Aspek keaslian meliputi: Tidak pernah dipublikasikan sebelumnya, Tidak ditemukan gejala kecurangan ilmiah, Tidak menyimpang dari norma akademik dan norma hukum sesuai dengan Peraturan Menteri Pendidikan Nasional No. 17 Tahun 2010 tentang Pencegahan dan Pananggulangan Plagiat di Perguruan Tinggi

3. Alur Validasi Karya Ilmiah Dosen

1. Dosen melakukan registrasi karya ilmiah di LPPM dan selanjutnya meng-upload karya tulisnya ke (<http://penelitian.unud.ac.id/simlit/>).
2. Dosen menyeminarkan karya ilmiah yang akan divalidasi di PS masing-masing.
3. Dosen mengirimkan karya ilmiah yang divalidasi melalui KPS kepada Dekan/Direktur Pascasarjana
4. Dekan/ Direktur Pascasarjana mengirimkan karya Ilmiah kepada Tim Validasi Fakultas/ Program Pascasarjana.

Tim Validasi Fakultas/Program Pascasarjana terdiri dari dua orang *reviewer* yaitu satu orang *reviewer* berasal dari PS yang sama dan satu orang *reviewer* dari PS

yang berbeda untuk Fakultas yang multi PS atau dua orang *reviewer* dari PS yang sama untuk Fakultas mono PS.

Reviewer melakukan pengecekan keaslian dan kelayakan dari karya ilmiah dosen dengan melakukan pelacakan secara manual dan online di Internet.

5. Karya ilmiah yang lolos validasi oleh Tim Validasi Fakultas/PPS dikirim oleh Dekan /Dir. Program Pascasarjana ke Rektor.

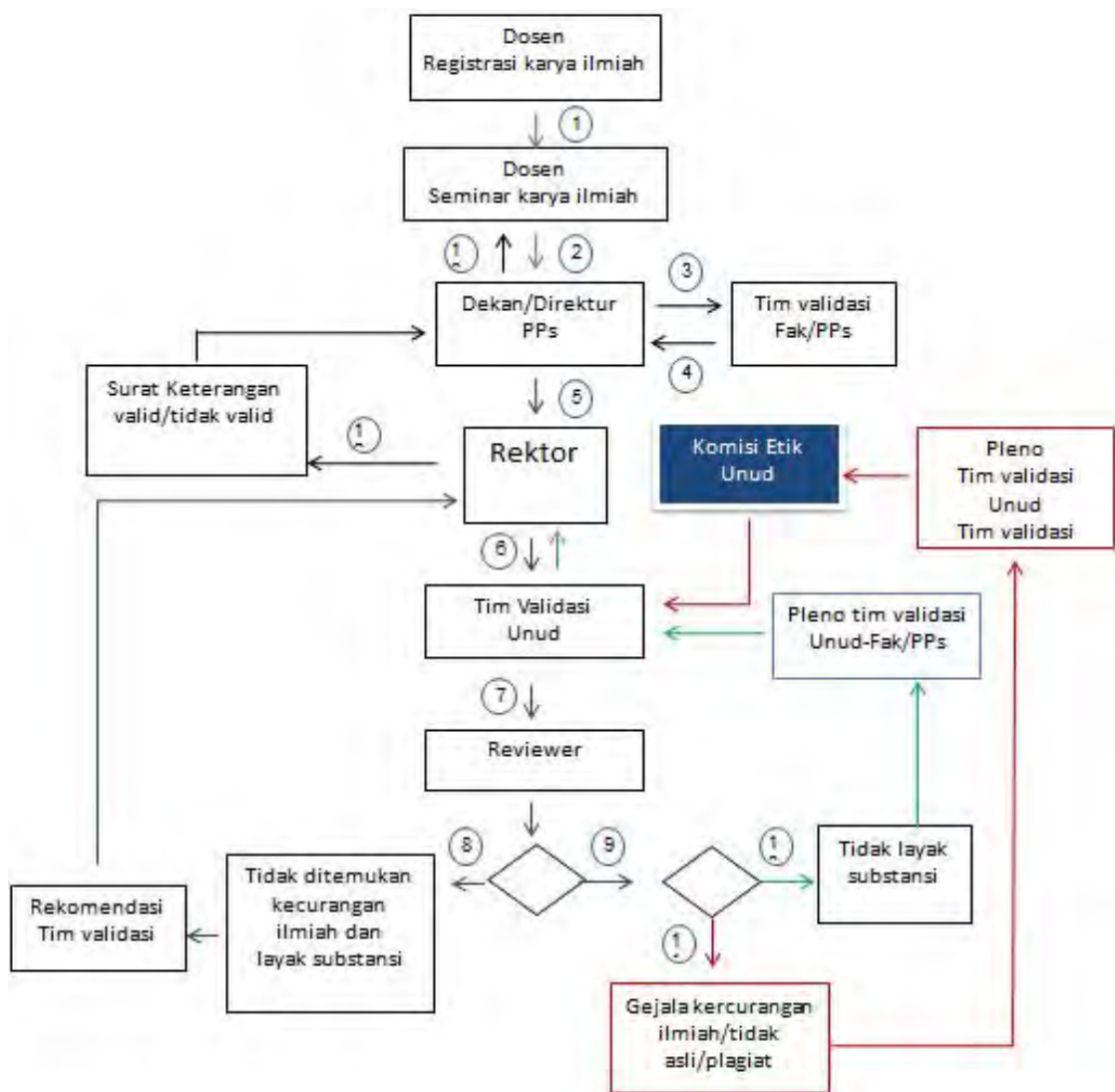
Jumlah karya tulis yang dikirim ke Tim Validasi adalah

Jenis	Jumlah <i>copy</i> yang disetor ke Tim Validasi Unud	File elektronik
Jurnal ilmiah dan prosiding	Foto copy (2 rangkap atau cetak lepas jurnal) , disertai dengan: copy cover depan jurnal, daftar mitra bestari, daftar isi, halaman karya tulis.	CD memuat file elektronik (pdf) setiap karya ilmiah yang divalidasi
Buku text, buku ajar, dan karya ilmiah lainnya selain jurnal ilmiah	Buku asli sebanyak 2 eksemplar disertai 2 buah foto copy cover depan, daftar isi buku, serta foto copy bagian yang memuat BAR CODE dan No. ISBN	CD memuat file elektronik (pdf) yang disetor

6. Rektor mengirimkan karya ilmiah yang divalidasi ke Ketua Tim Validasi Karya Ilmiah tingkat Universitas (Pembantu Rektor Bidang Akademik). Sekretariat Tim Validasi Karya Ilmiah Unud akan mengirimkan karya ilmiah yang divalidasi kepada *reviewer* yang terdiri dari dua orang *reviewer* yaitu satu orang *reviewer* berasal dari Fakultas yang sama dan satu orang *reviewer* dari Fakultas yang berbeda yang bidang ilmunya serumpun.
7. Tim Validasi Karya Ilmiah Unud akan melakukan penelaahan kelayakan dan keaslian karya-karya ilmiah dengan melakukan pelacakan secara manual dan online di Internet.
8. Apabila tidak ditemukan kecurangan ilmiah serta karya Ilmiah secara substantif dianggap layak, maka Tim Validasi Karya Ilmiah Unud akan mengirimkan Rekomendasi kepada Rektor dan Tim IT. Selanjutnya Tim IT mencetak surat keterangan valid (langkah no.12)
9. Apabila secara substansi kurang layak atau ditemukan masalah-masalah lain selain kecurangan ilmiah, maka dilanjutkan ke langkah no. 10 dan 11
10. Dilakukan sidang pleno Tim Validasi Karya Ilmiah Unud serta Tim Validasi Karya Ilmiah Fakultas. Apabila setelah sidang pleno masih ada masalah kelayakan substansi dan keaslian, maka Tim Validasi Karya Ilmiah Unud akan mengirimkan

rekomendasi ke Rektor dan Tim IT. Selanjutnya Tim IT mencetak surat keterangan valid/tidak valid

11. Apabila ditemukan indikasi kecurangan ilmiah, ketidakaslian dan indikasi plagiat, maka akan dilakukan sidang pleno oleh Tim Validasi Karya Ilmiah Unud dengan mengundang Tim Validasi Karya Ilmiah Fakultas. Apabila diputuskan adanya indikasi kecurangan ilmiah maka akan dilakukan penanggulangan seperti dalam SOP pencegahan dan penanggulangan plagiarsisme Unud oleh Komisi Etik Universitas Udayana. Hasil pleno ini dilaporkan kepada Rektor dan Tim IT oleh Tim Validasi Karya Ilmiah Unud. Selanjutnya Tim IT mencetak surat keterangan “tidak ditemukan/ ditemukan” ketidakjujuran ilmiah.



Gambar 2.4. Diagram alir validasi karya ilmiah di lingkungan Unud

12. Rektor memberikan Surat Keterangan “*Valid/tidak valid*” terhadap karya ilmiah yang diusulkan.
13. Surat Keterangan Valid/tidak valid dikirim ke pada Dekan /Direktur PPS yang selanjutnya oleh Dekan/Direktur PPS diteruskan ke pada pengusul.

BAB 3

TUJUAN DAN MANFAAT PENELITIAN

3.1 Tujuan Khusus

Penelitian ini bertujuan:

1. Membuat sebuah sistem media penyimpanan terdistribusi untuk menangani pertumbuhan data validasi karya ilmiah menggunakan metode cloud storage
2. Membuat sebuah sistem validasi yang mampu melakukan pemeriksaan plagiarisme secara langsung
3. Mampu melakukan sistem validasi secara reviewer based dengan mengembangkan aplikasi native smartphone khususnya android.

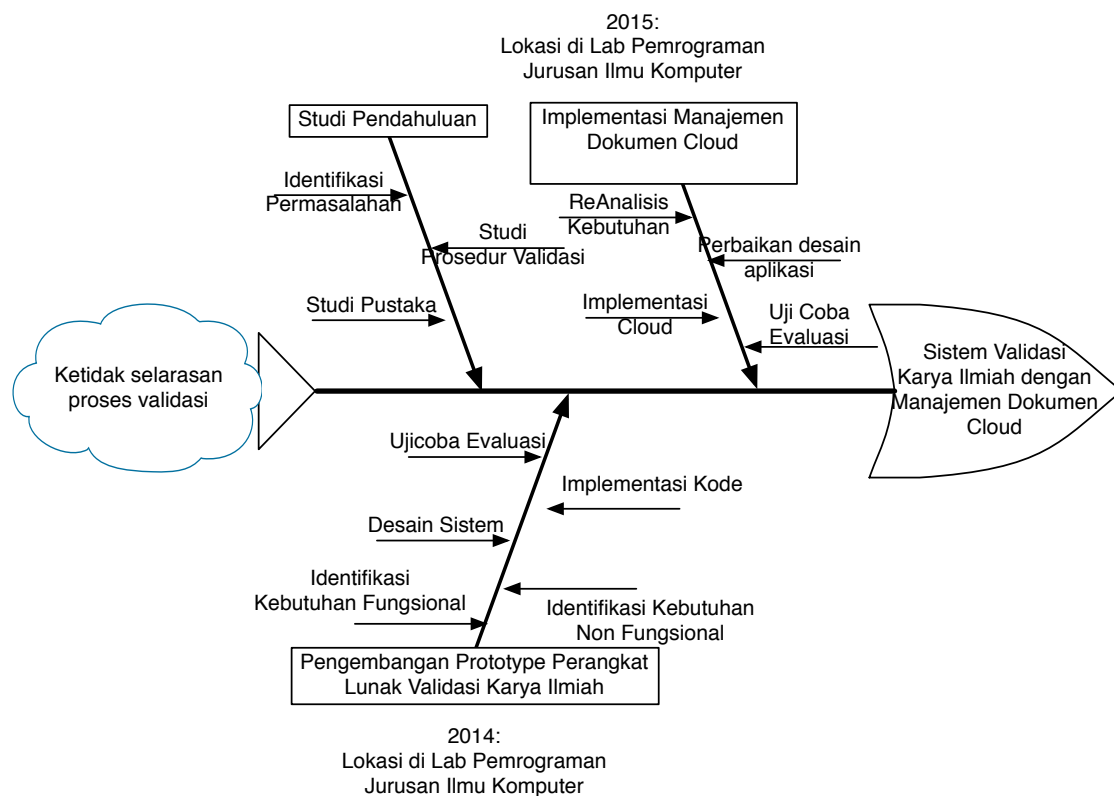
BAB 4 METODE PENELITIAN

4.1. Waktu dan Tempat

Penelitian ini akan dilakukan selama dua tahun di laboratorium pemrograman Jurusan Ilmu Komputer Universitas Udayana dengan mengambil studi kasus proses validasi karya ilmiah yang dikelola dan dikoordinasikan oleh BPMU dan Lembaga Penelitian Universitas Udayana.

4.2. Tahapan Penelitian

Secara garis besar penelitian ini dibagi menjadi dua tahapan (dua tahun) seperti yang dapat dilihat pada gambar 3.1. yang direpresentasikan dengan menggunakan *fishbone Diagram*.



Gambar 3.1. Skema Alir Penelitian

Berdasarkan skema alir penelitian pada gambar 3.1 dapat dilihat bahwa permasalahan utama yang menjadi dasar penelitian ini adalah karena adanya ketidakselarasan proses validasi karya ilmiah. Untuk mengatasi permasalahan tersebut atau tercapainya sistem validasi karya ilmiah dengan manajemen dokumen cloud diperlukan beberapa tahap pengembangan

sistem yaitu: studi pendahuluan yang bertujuan untuk mendapatkan kebutuhan sistem yang hendak dibangun, kemudian dilanjutkan dengan mengembangkan prototype perangkat lunak validasi yang standar, lalu setelah prototype selesai dibuat, dilanjutkan dengan mengimplementasikan sistem *cloud document* yang dapat mendukung penyimpanan dokumen validasi dalam ukuran sangat besar. Apabila semua tahap telah terimplemntasikan dengan baik, maka dapat dianggap sistem validasi karya ilmiah dengan manajemen dokumen cloud dapat tercapai.

Berikut uraian lebih lanjut mengenai setiap tahapan penelitian yang ditampilkan pada gambar

4.2.1. Tahapan Awal

Pada tahap ini, maksud dan tujuan dari penelitian ini dielaborasi dan didefinisikan lebih lanjut berkaitan dengan validasi karya ilmiah khususnya di Universitas Udayana. Peneliti mempelajari dan mengevaluasi prosedur dari proses validasi karya ilmiah yang diterapkan serta yang dialami selama melakukan proses tersebut pada tahun-tahun sebelumnya. Pada tahap ini juga dilakukan studi pustaka awal mengenai topik yang diteliti untuk mendapatkan informasi sejauh mana penelitian yang sejenis pernah dilakukan.

4.2.2. Tahapan Pertama (tahun I)

Tahap pertama dari penelitian ini adalah tahap pengembangan sistem aplikasi pengelolaan proses validasi karya ilmiah. Pada tahap ini, sistem akan dibangun dan ditempatkan di lingkungan web server tunggal, dimana tujuan utama pada tahap ini adalah fokus pada proses realisasi sistem aplikasi validasi karya ilmiah. Untuk mendukung pelaksanaan tahap pertama ini, penelitian dibagi menjadi beberapa sub-tahapan, dimana merupakan bagian dari metodologi pengembangan perangkat lunak SDLC (*Software Development Life Cycle*).

Setelah didapatkan output dari penelitian tahun pertama yaitu berupa prototype validasi karya ilmiah yang sudah berjalan sesuai dengan kebutuhan user. Maka sistem ini perlu dilanjutkan untuk meningkatkan kemampuannya. Adapun peningkatan kemampuan yang diharapkan dipaparkan pada sub bab berikutnya.

4.2.3 Tahap Kedua (tahun II)

Seperti yang telah diuraikan di latar belakang bahwa proses validasi karya ilmiah akan melibatkan begitu banyak dokumen digital, terutama dokumen dengan format *portable document format* (pdf), sehingga dibutuhkan mekanisme/desain yang tepat sehingga kinerja

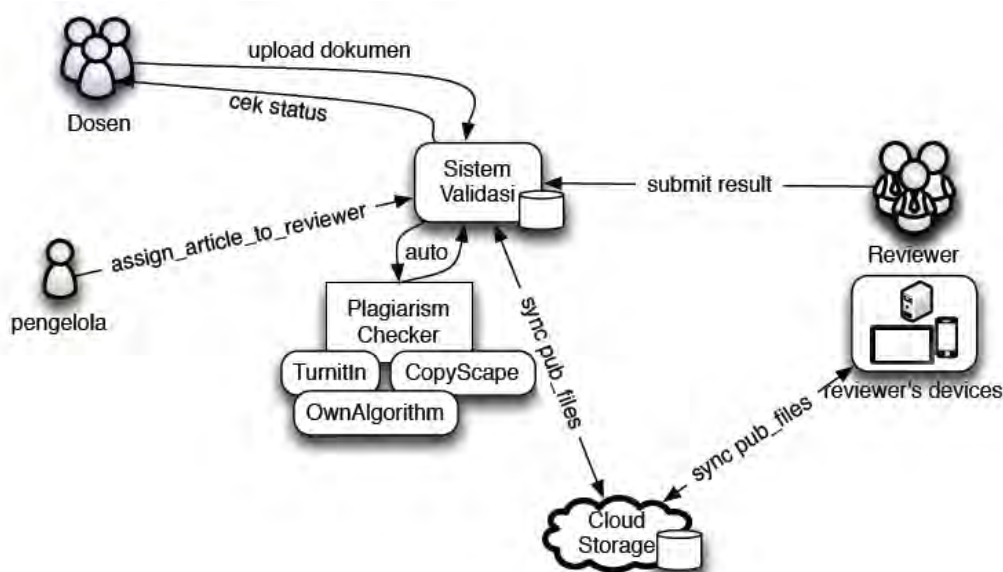
dari sistem dapat tetap terjaga dengan baik. Selain itu, bagaimana tingkat keamanan dari sistem sehingga arsip digital yang tersimpan tidak mudah hilang. Kebutuhan-kebutuhan ini merupakan salah satu bagian dari Kebutuhan Non-Fungsional yang nantinya harus dimiliki oleh sistem. Oleh karena itu, pada tahap kedua ini, penelitian akan difokuskan pada pemenuhan kebutuhan non-fungsional serta bagaimana penyesuaiannya dengan sistem yang telah dikembangkan pada tahap penelitian sebelumnya. Pada tahap ini akan dieksplorasi lebih lanjut teknologi *Cloud Computing* dan *Cloud Storage* dengan melakukan eksperimen.

1. Analisis Hasil Ujicoba tahap pertama

Pada tahap ini dilakukan analisis hasil ujicoba sistem aplikasi yang telah dikembangkan pada tahap sebelumnya.

2. Penyesuaian desain dan arsitektur sistem

Pada tahap ini, hasil analisis kemudian digunakan untuk melakukan penyesuaian desain dari perangkat lunak. Karena pada tahap kedua ini difokuskan pada pemenuhan kebutuhan non-fungsional, maka pada tahap ini juga dilakukan penyesuaian arsitektur sistem secara keseluruhan, yakni *Cloud Storage* sebagai teknologi pilihan dalam menghadapi pertumbuhan data digital serta kinerja akses sistem. Disamping itu pula, akan dikembangkan *plagiarism checker* yang diharapkan mampu memberikan kemudahan bagi reviewer untuk melakukan pemeriksaan kemiripan artikel ilmiah. Desain arsitektur yang nantinya kemungkinan digunakan dapat dilihat seperti pada gambar 3.3



Gambar 3.3. Desain arsitektur sistem tahap II

3. Penyesuaian implementasi kode dan sistem

Pada tahap ini, dilakukan proses perbaikan atas perubahan desain perangkat lunak maupun perbaikan karena ketidak sempurnaan implementasi pada tahap sebelumnya. Selain itu pada tahap ini juga dilakukan implementasi arsitektur sistem cloud serta mengiterkoneksi perangkat lunak ke *cloud storage*.

4. Ujicoba dan Evaluasi Sistem

Pada tahap ini, kembali dilakukan ujicoba perangkat lunak. Sama seperti pada tahun sebelumnya, ujicoba dilakukan dengan menguji semua fitur yang dimiliki oleh perangkat lunak dengan metode *blackbox* dan skenario yang sama. Hanya saja pada uji ini, dilakukan pengujian tambahan yakni *web-stress test*, yakni proses simulasi akses sistem yang dilakukan secara bersamaan dimana jumlah client yang melakukan akses dapat dimanipulasi sesuai dengan desain pengujiannya. Hal ini juga dilakukan untuk menguji performa dari sistem setelah dipergunakan pendekatan cloud storage.

4.3 Indikator Capaian

Berikut pada Tabel 3.1 disajikan Indikator capaian dari penelitian ini.

Tabel 3.1 Indikator capaian penelitian

Tahun	Kegiatan	Indikator Capaian	Luaran
2015	1. Analisis pertama	1. Penyesuaian Fungsional dan Non Fungsional	1. Sistem Ilmiah dengan <i>cloud computing</i> dan <i>cloud storage</i>
	2. Penyesuaian desain dan arsitektur sistem	2. Perbaikan Arsitektur	2. Diseminasi Internasional
	3. Implementasi Cloud Storage	3. Kebutuhan Fungsional dan Non Fungsional (<i>Cloud Storage</i>) terimplementasi dengan baik	3. Publikasi di Jurnal Nasional Terakreditasi
	4. Ujicoba dan Evaluasi		

BAB 5

HASIL YANG DICAPAI

5.1 Analisa Kebutuhan

Pada tahap analisa kebutuhan sistem dilakukan wawancara dengan pihak pengguna, dalam hal ini BPMU dan reviewer. Berdasarkan hasil wawancara dengan BPMU didapatkan kebutuhan sebagai berikut:

5.1.1 Kebutuhan Fungsional

- i. Sistem mampu dijalankan secara online (*web based*)
- ii. Sistem mampu melakukan pendelegasian reviewer pada suatu publikasi ilmiah
- iii. Reviewer mampu memberikan nilai secara online
- iv. Dosen dapat mengupload karya ilmiahnya secara online
- v. Sistem diharapkan dapat melakukan manajemen atau *Create, Read, Update, Delete* yang disingkat CRUD terhadap jenis penelitian. Jenis penelitian yang ada saat ini adalah: Dosen Muda, Hibah Udayana, Dasar, Kajian Wanita, Hibah Bersaing, Hibah Pascasarjana, Hibah Pekerti, Hibah Kompetensi, Insentif Dasar, Hibah Strategis Nasional, Kerjasama, Fundamental, dan Hibah Kerjasama.
- vi. Sistem diharapkan dapat melakukan manajemen data jenis publikasi. Jenis publikasi yang ada saat ini adalah: Majalah Populer/Koran, Seminar Nasional, Seminar Internasional, Prosiding(ISBN), Jurnal Nasional Belum Akreditasi, Jurnal Nasional Terakreditasi, Jurnal Internasional, dan Buku.
- vii. Sistem diharapkan mampu melakukan manajemen terhadap kategori publikasi. Local, nasional, atau internasional.
- viii. Sistem dapat mendata jumlah penulis dari tiap karya ilmiah.
- ix. Sistem mampu melakukan: pelacakan history komentar pada tiap karya ilmiah
- x. Sistem mampu melakukan penilaian terhadap karya ilmiah.

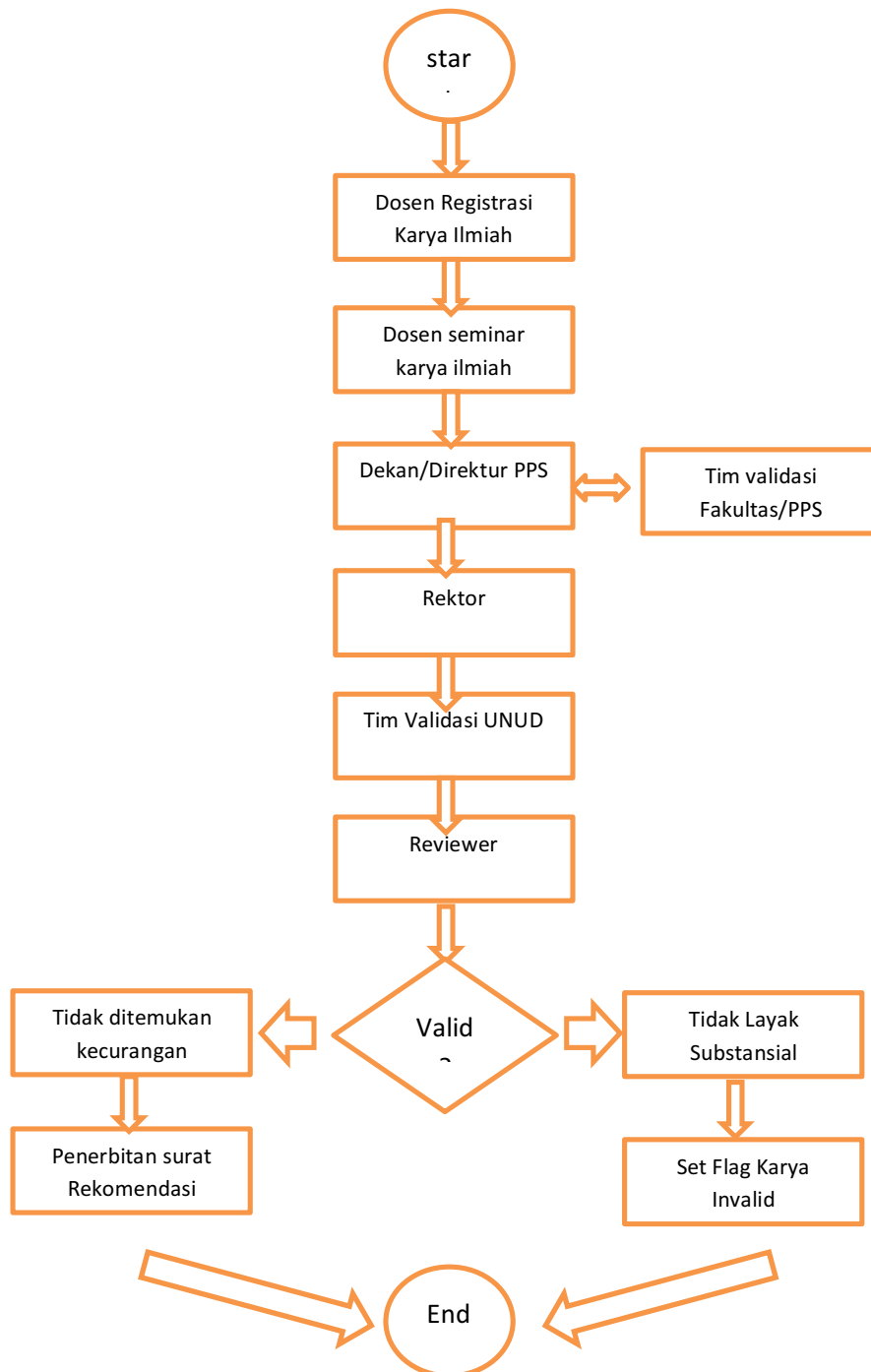
5.1.2 Kebutuhan Non Fungsional

- i. Sistem mampu melakukan manajemen berita, pengumuman dan panduan
- Kebutuhan sistem dipandang dari reviewer masih belum didapatkan. Sembari merampungkan sistem yang diinginkan oleh user BPMU, akan dilakukan pula survey

tambahan kepada para reviewer untuk mencari model review yang sekiranya dapat meningkatkan kenyamanan reviewer dalam melakukan proses review.

5.2 Flow Sistem

Setelah kebutuhan fungsional dan non fungsional sistem didapat, langkah selanjutnya adalah merancang flow sistem.



Flowchar 1. Flowchar sistem validasi

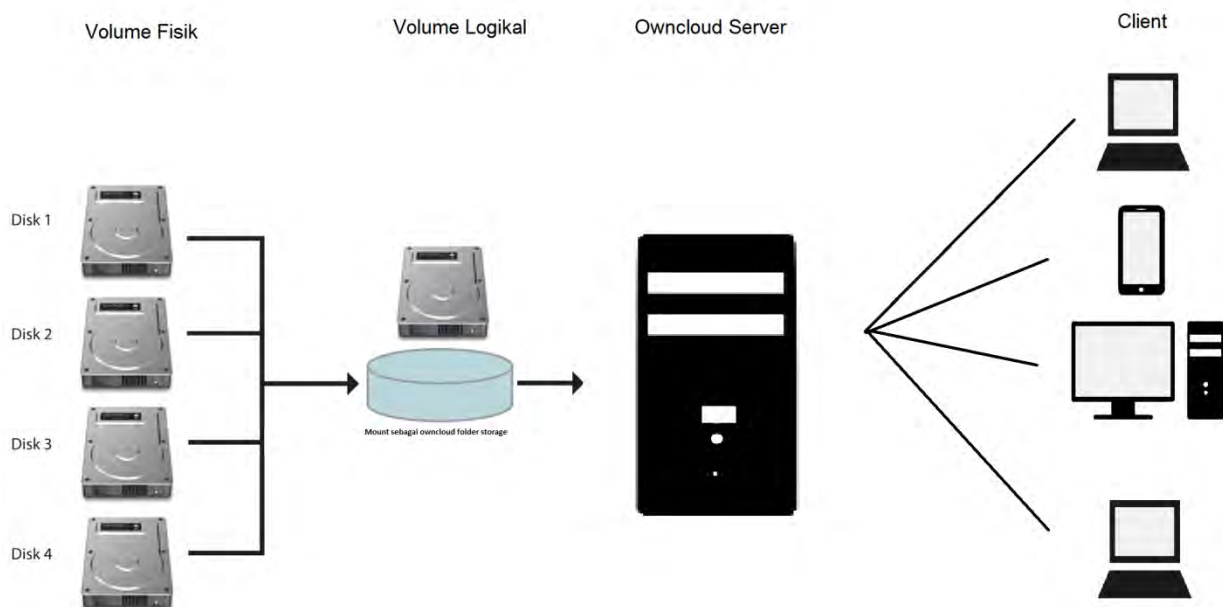
Setelah sistem validasi berbasis web rampung, maka selanjutnya akan dilakukan pengembangan pada manajemen data terpusat. Penyimpanan data yang terpusat berarti sumber data dapat dari berbagai macam sumber, kontributor, sehingga setiap file user dapat dikelola dengan mudah, reliable dan dinamis. Sehingga pertukaran data dapat dilakukan dengan mudah, cepat dan praktis. Ditambah lagi dengan sistem ini tidak membutuhkan koneksi Internet untuk mengaksesnya sehingga akses user tidak membutuhkan bandwidth internet dan dana tambahan untuk membeli pulsa modem. Untuk itu sistem cloud yang akan dikembangkan adalah cloud (komputasi awan) IaaS (Infrastructure as a Service). Namun apakah cloud/komputasi awan itu ? menurut Wikipedia, Komputasi awan (bahasa Inggris: cloud computing) adalah gabungan pemanfaatan teknologi komputer ('komputasi') dan pengembangan berbasis Internet ('awan'). Awan (cloud) adalah metafora dari internet, sebagaimana awan yang sering digambarkan di diagram jaringan komputer.

Nama besar, seperti IBM, Microsoft, Google, dan Apple adalah beberapa contoh penguasa terbesar komputasi awan. IBM misalnya pada akhir tahun 2009 meluncurkan LotusLive, layanan kolaborasi berbasis cloud, Microsoft, Ray Ozzie sebagai chief software architect pengganti Bill Gates, sudah mengadopsi windows Azure, sistem operasi berbasis cloud menjadi masa depan Windows OS. Apple menyediakan layanan Mobile Me yang memungkinkan pengguna produk Mac melakukan sinkronisasi data dalam cloud. Sementara google memberikan layanan google docs. Layanan ini memungkinkan user membuat dokumen secara online tanpa perlu menginstall software di PC atau notebook. Google juga meluncurkan sistem operasi cloudnya yaitu sistem operasi alternative dari sistem operasi yang sudah adayang kemungkinan besar menjadi ancaman serius bagi penyedia sistem operasi.

Sebagai contoh dalam google drive ini, memungkinkan setiap user dapat memiliki kapasitas free 5GB (versi free) sehingga dapat diletakkan berbagai macam file, di share dan hebatnya memungkinkan penggunaanya untuk menggunakan file ini secara kolaboratif/bersama-sama dan real time. Pastinya semuanya free, hanya pastikan anda memiliki akun gmail. solusi opensource untuk hal ini dapat menggunakan ownCloud.

5.3 Desain dan Implementasi

untuk dapat mengimplemntasikan layanan owncloud, ada beberapa hal yang harus dipahami terlebih dahulu. Yaitu Logical Volume Management (LVM). LVM (<http://sources.redhat.com/lvm>) adalah Logical Volume Management atau sebuah manajemen penyimpan di sistem operasi GNU/Linux yang menyediakan fleksibilitas dalam membuat dan mengubah partisi dalam sebuah disk. LVM menambahkan lapisan antara antarmuka I/O kernel dengan perangkat fisik untuk mendapatkan logicalview dari penyimpan. Dengan adanya lapisan inilah kita dapat melakukan perubahan partisi baik itu resize, penghapusan, dan lainlain secara live pada sistem baik untuk satu atau lebih media. Kemudahan tersebut akan membuat administrator maupun pengguna lebih nyaman dalam melakukan manajemen dan alokasi ruang penyimpan hingga penamaan volume.



Gambar 5.1 menunjukkan topologi yang digunakan untuk owncloud server.

Dapat terlihat dimana terdapat beberapa fisik harddisk yang tersedia kemudian digabungkan pada gambar 5.1, dimana menjadi satu kesatuan dan di lakukan mounting kedalam folder yang dimana folder ini akan menjadi folder storage untuk owncloud server menyimpan data dari setiap client. Model seperti inilah yang nantinya akan dibangun sebagai pendukung sistem validasi karya ilmiah berbasis cloud

5.3.1 Fitur Sistem

Fitur yang akan dikembangkan dalam sistem validasi karya ilmiah berbasis cloud ini adalah:

1. Sistem nantinya akan dapat menambahkan bebarapa storage secara dinamis, tanpa diketahui oleh user secara langsung sehingga kebutuhan user akan storage dapat selalu terpenuhi.
2. Sistem dapat diakses melalui sistem berbasis web dengan menggunakan API dari owncloud yang diimplementasikan pada bahasa pemrograman PHP

5.3.2 Implementasi

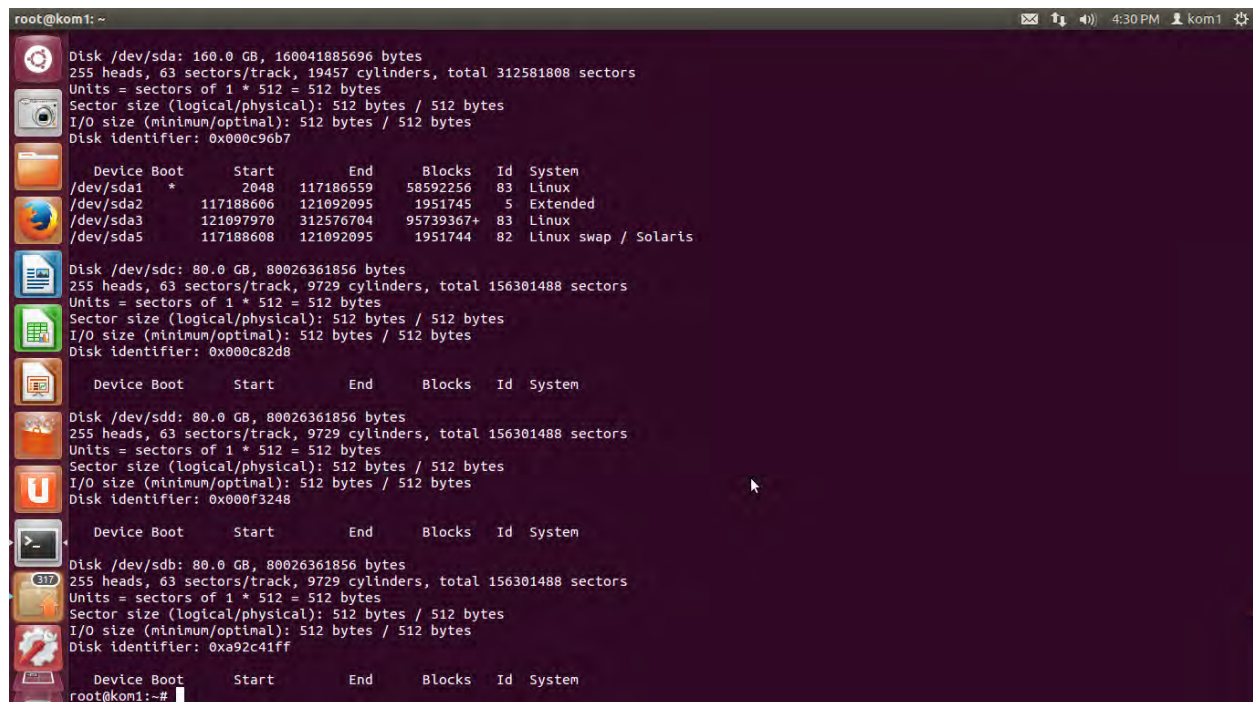
implementasi sistem yang akan dibangun, dibedakan menjadi 2 tahap yaitu pengembangan server owncloud yang dinamis dan pengembangan client owncloud yang terintegrasi dengan sistem validasi karya ilmiah di Universitas Udayana.

5.3.2.1 Persiapan pengembangan Sistem Owncloud

Tahap pertama yaitu perlu diperiksa terlebih dahulu harddik/logical volume yang tersedia pada computer server owncloud dengan perintah sebagai berikut :

```
sudo fdisk -l
```

Maka akan terlihat seperti gambar 5.2 dibawah ini :



Gambar 2 Daftar Harddisk Yang Terinstall

Terlihat seperti gambar 5.2 bahwa terdapat 4 buah harddisk yaitu masing-masing 160gb (tempat system operasi ubuntu dan owncloud di install), 80gb (harddisk 2), 80gb (harddisk 3), 80gb (harddisk 4). Dari keempat harddisk ini yang nantinya akan kita gabungkan menjadi seolah-olah satu harddisk secara logical dan akan dimount sebagai folder tempat data dari setiap user owncloud di simpan.

Langkah selanjutnya adalah menggabungkan drive/harddisk, yaitu dengan perintah sebagai berikut :

```
Pvcreate dev/partisi_harddisk
```

Dapat terlihat seperti gambar 5.3

```
root@kom1:~# pvcreate /dev/sda3
Physical volume "/dev/sda3" successfully created
root@kom1:~# █
```

Gambar 5.3 Tahap Menggabungkan Harddisk

Lakukan hingga semua partisi harddisk tergabung

```
root@kom1:~# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
root@kom1:~# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created
root@kom1:~# pvcreate /dev/sdd
Physical volume "/dev/sdd" successfully created
root@kom1:~# █
```

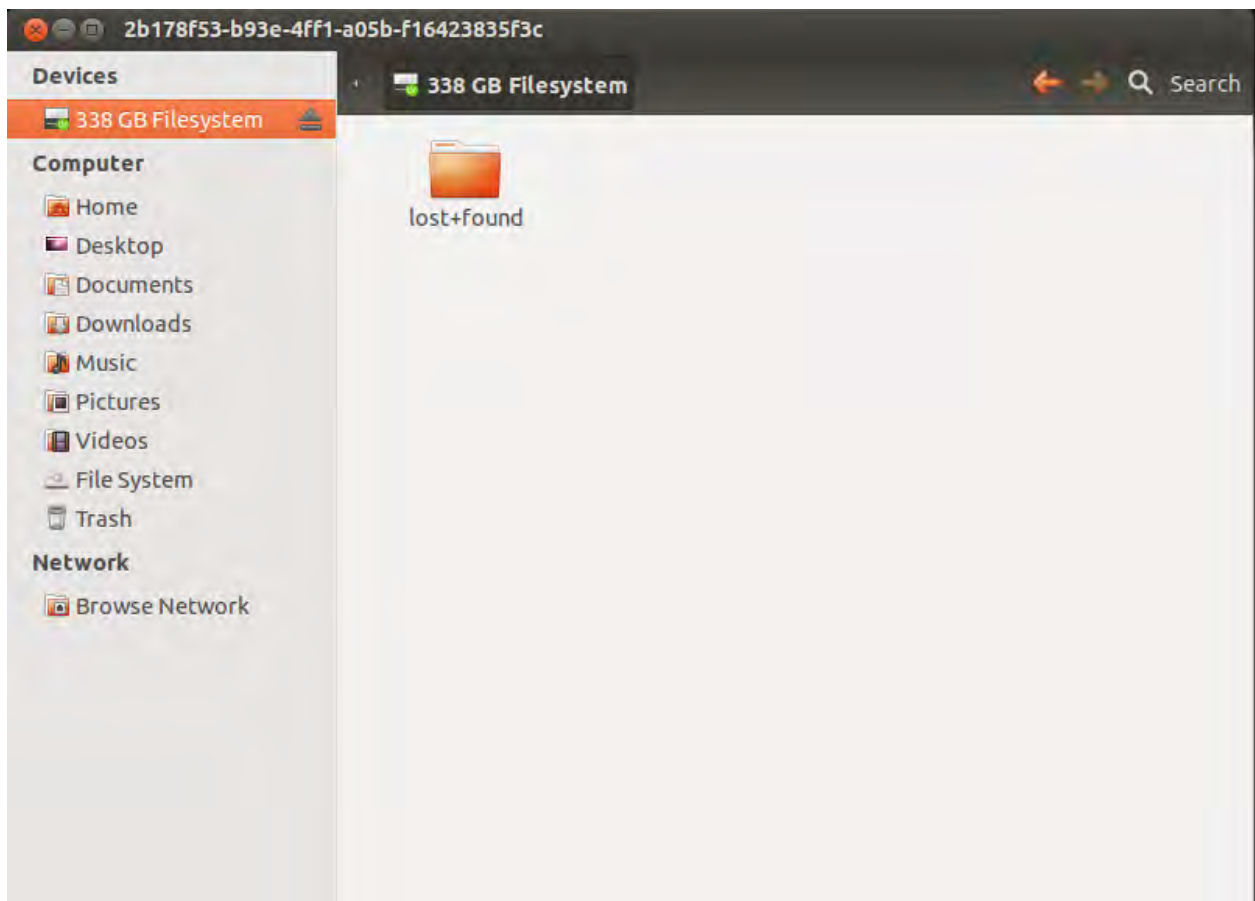
Gambar 5.4 Semua Harddisk Telah digabungkan

Setelah semua partisi tergabung maka akan terlihat seperti gambar 5.5

```
root@kom1:~# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sda3   vg-own lvm2 a-   91.30g 91.30g
/dev/sdb    vg-own lvm2 a-   74.53g 74.53g
/dev/sdc    vg-own lvm2 a-   74.53g 74.53g
/dev/sdd    vg-own lvm2 a-   74.53g 74.53g
root@kom1:~# vgcreate vg-owncloud /dev/sda3 /dev/sdb /dev/sdc /dev/sdd
Volume group "vg-owncloud" successfully created
root@kom1:~# vgs
VG          #PV #LV #SN Attr   VSize   VFree
vg-owncloud  4   0   0 wz--n- 314.88g 314.88g
root@kom1:~#
```

Gambar 5.5 Penggabungan Haddisk Selesai

Terlihat seperti gambar 5.5 bahwa hariddisk yang telah digabungkan telah menjadi satu kesatuan volume logical.



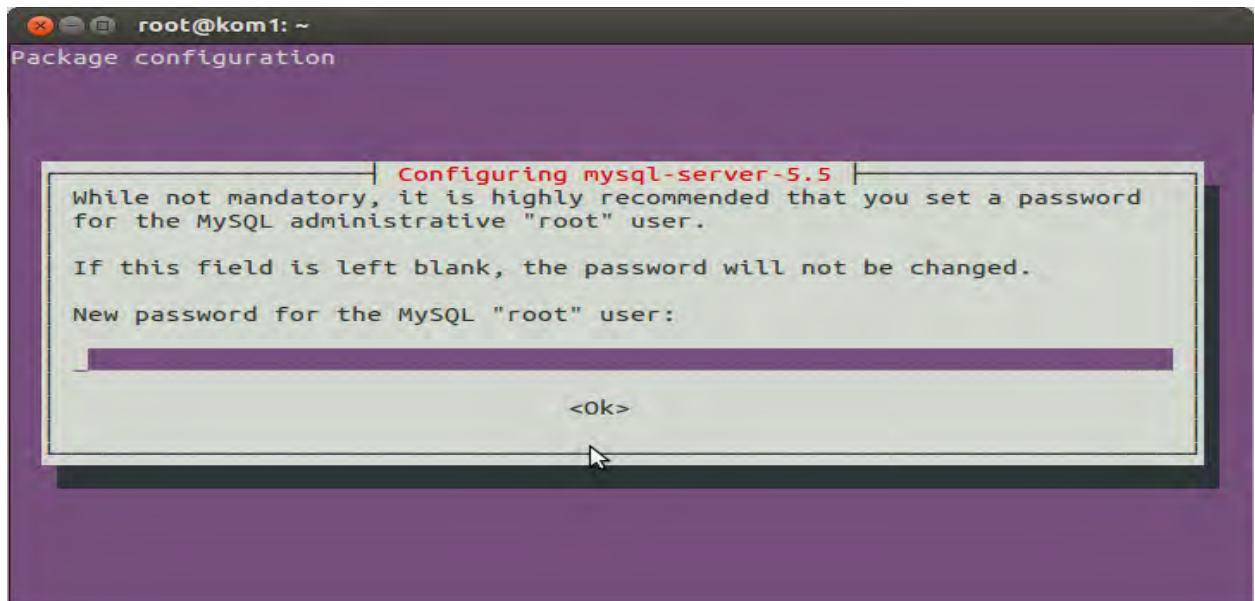
Gambar 5.6 Drive Virtual Haddisk

5.3.2.2 Langkah-langkah instalasi Owncloud dan konfigurasi yaitu :

Langkah awal yaitu Install terlebih dahulu webserver pada server yang telah kita konfigurasi LVM.

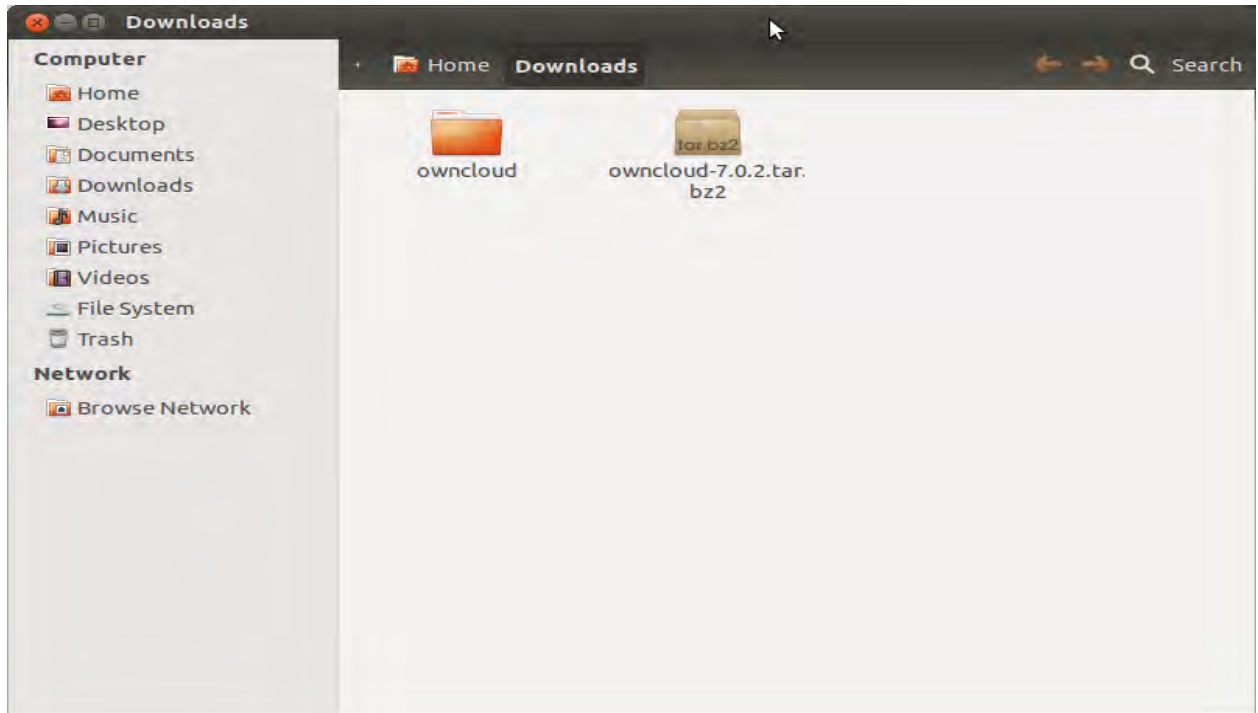
```
$ sudo apt-get install apache2 php5 mysql-server phpmyadmin
```

Dalam instalasi apache dan mysql ini nantinya akan diminta password dan konfigurasi lainnya seperti gambar 5.7



Gambar 7 Form Password My-sql

Konfigurasi dan berikan password pada setiap form yang ditampilkan. Setelah konfigurasi mysql dan apache selesai maka yang kita lakukan selanjutnya yaitu : mendownload file owncloud yang nantinya akan kita tempatkan pada folder web server. Untuk dapat mendownload file owncloud dapat di unduh pada website resminya yaitu www.owncloud.com

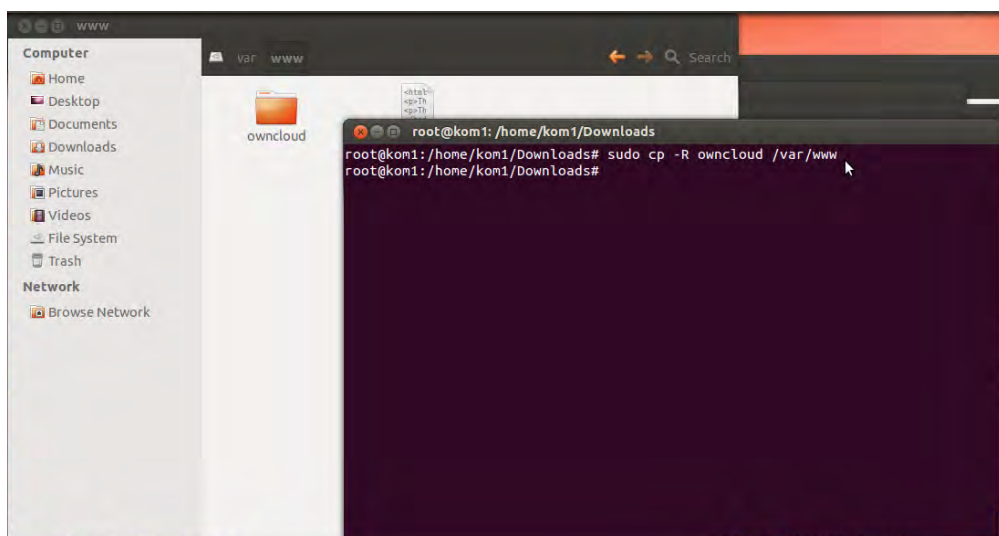


Gambar 5.8 Ekstrasi File Owncloud

Setelah selesai mengunduh maka tahap selanjutnya yaitu extrack file owncloud seperti gambar 5.8. Langkah selanjutnya yaitu copy folder owncloud tersebut kedalam folder webserver. Dalam penelitian kali ini menggunakan folder web server `/var/www` dengan perintah :

```
sudo cp -R owncloud /var/www
```

Maka akan terlihat seperti gambar 5.9 dibawah ini :



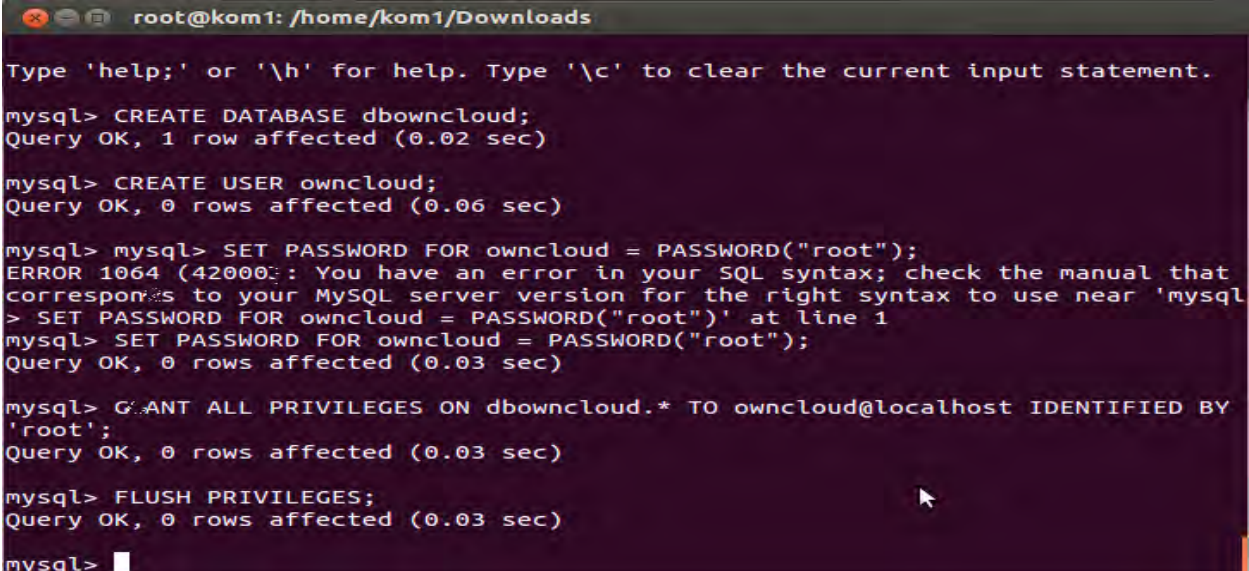
Gambar 5.9 Direktori Owncloud

Langkah selanjutnya berikan hak kepemilikan direktori tersebut menjadi milik user *www-data* dengan perintah :

```
sudo chown -R www-data.www-data /var/www/owncloud
```

Selanjutnya yaitu membuat database sekaligus user khusus untuk menghandle aplikasi owncloud. Login terlebih dahulu ke Mysql :

```
$ mysql -u root -p
mysql> CREATE DATABASE dbowncloud;
mysql> CREATE USER owncloud;
mysql> SET PASSWORD FOR owncloud = PASSWORD("root");
mysql> GRANT ALL PRIVILEGES ON dbowncloud.* TO owncloud@localhost
IDENTIFIED BY 'root';
mysql> FLUSH PRIVILEGES;
```



```
root@kom1: /home/kom1/Downloads
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> CREATE DATABASE dbowncloud;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER owncloud;
Query OK, 0 rows affected (0.06 sec)

mysql> mysql> SET PASSWORD FOR owncloud = PASSWORD("root");
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'mysql
> SET PASSWORD FOR owncloud = PASSWORD("root")' at line 1
mysql> SET PASSWORD FOR owncloud = PASSWORD("root");
Query OK, 0 rows affected (0.03 sec)

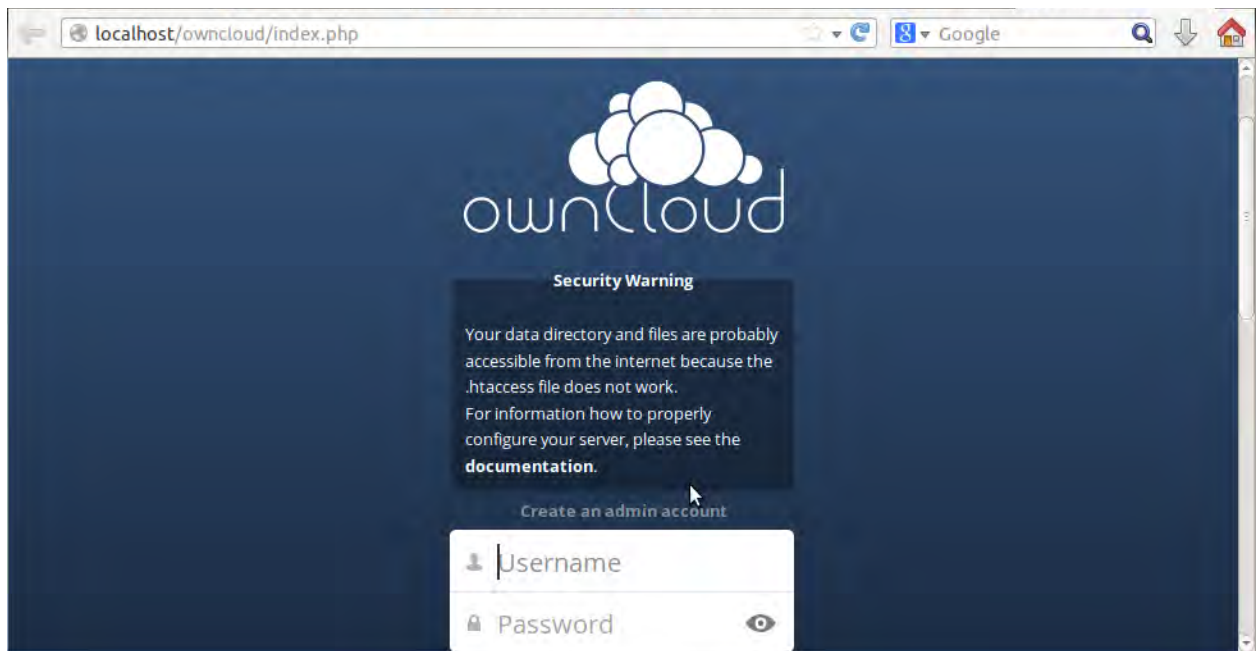
mysql> GRANT ALL PRIVILEGES ON dbowncloud.* TO owncloud@localhost IDENTIFIED BY
'root';
Query OK, 0 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

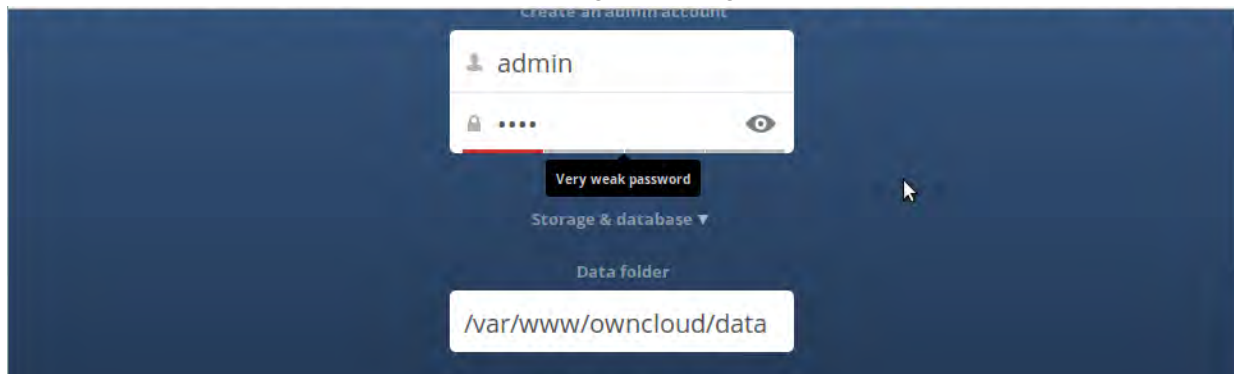
Gambar 5.10 Konfigurasi My-sql

Jika telah melakukan konfigurasi pada mysql seperti gambar 10 ,maka selanjutnya kita dapat melakukan konfigurasi pada owncloud. Lakukan konfigurasi pada owncloud dengan membuka terlebih dahulu owncloud yang telah terinstall pada koputer server dengan mengakses folder owncloud pada halaman web browser seperti pada gambar 5.11

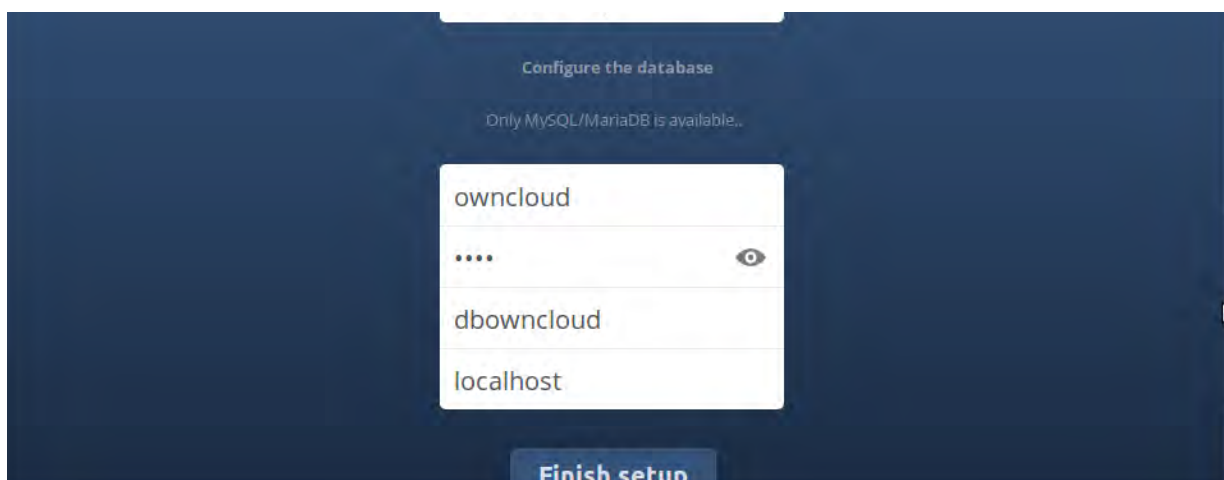


Gambar 5.11 Akses Owncloud Pada Webbrowser

Lalu lakukan konfigurasi sebagai berikut

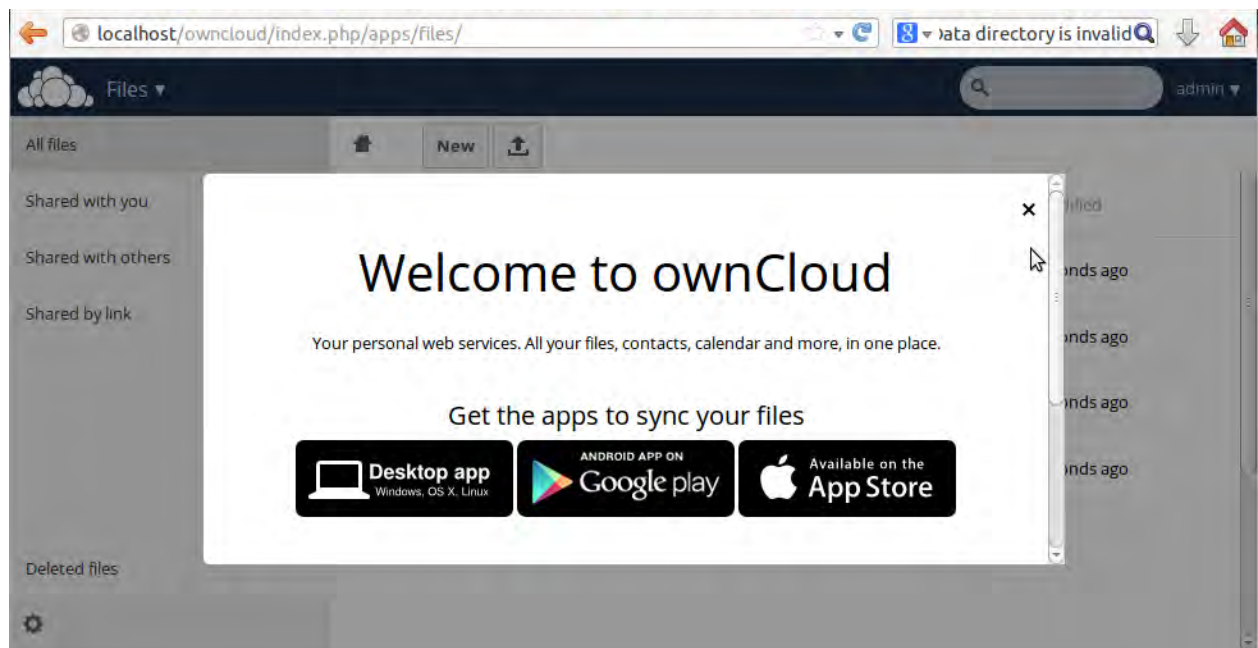


Gambar 5.12 Data Folder Default Owncloud



Gambar 5.13 Konfigurasi Database Owncloud

Setelah konfigurasi yang dilakukan sudah sesuai dengan database yang telah kita buat terlebih dahulu tadi, maka owncloud akan selesai dikonfigurasi dan akan tampil seperti gambar 5.14

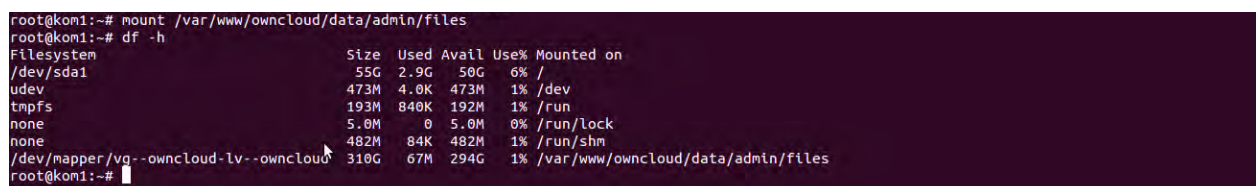


Gambar 5.14 Owncloud Selesai Di Konfigurasi

Selanjutnya perlu dilakukan konfigurasi yaitu LVM agar kita mount di direktori default owncloud yaitu `var/www/owncloud/data`, tetapi dalam penelitian ini direktori LVM akan di mount hanya untuk user admin sebagai penelitian, dan user admin ini akan diakses pada owncloud client dan akan dilakukan percobaan upload data dari client yang nantinya akan tersinkronasi pada sisi server.

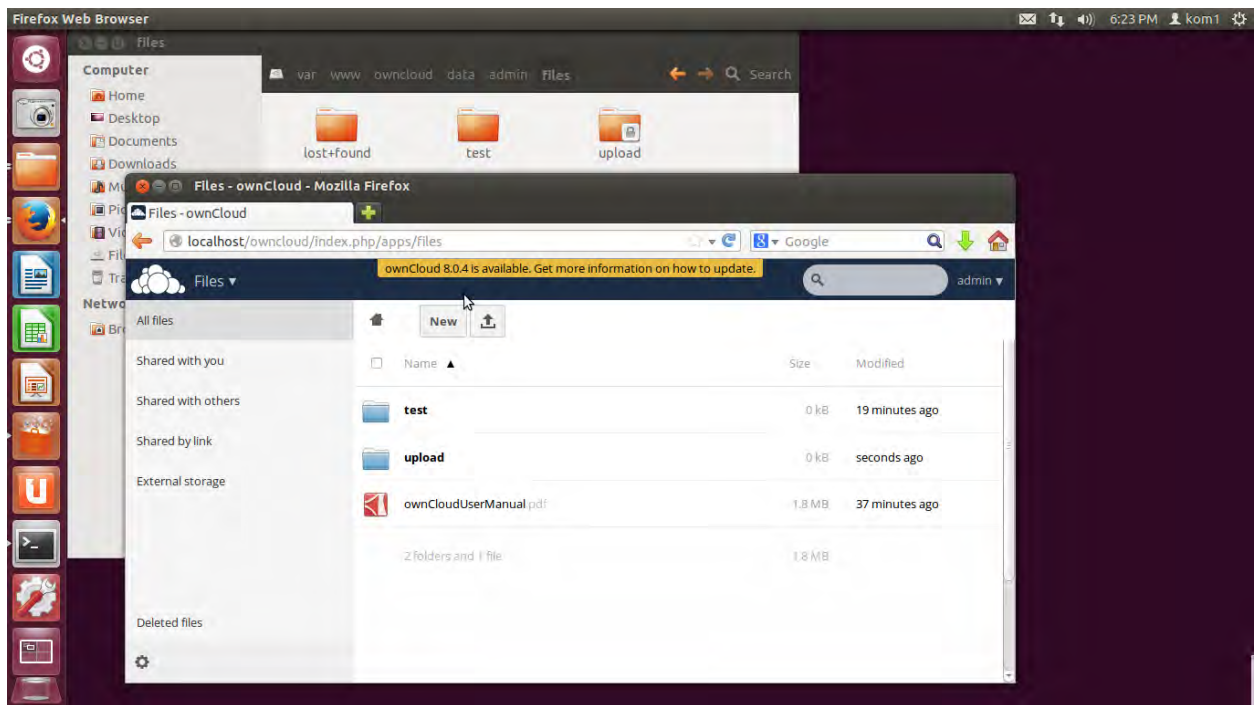
Langkah untuk melakukan mounting pada direktori admin owncloud yaitu :

```
$ Mount /var/www/owncloud/data/admin/files
```

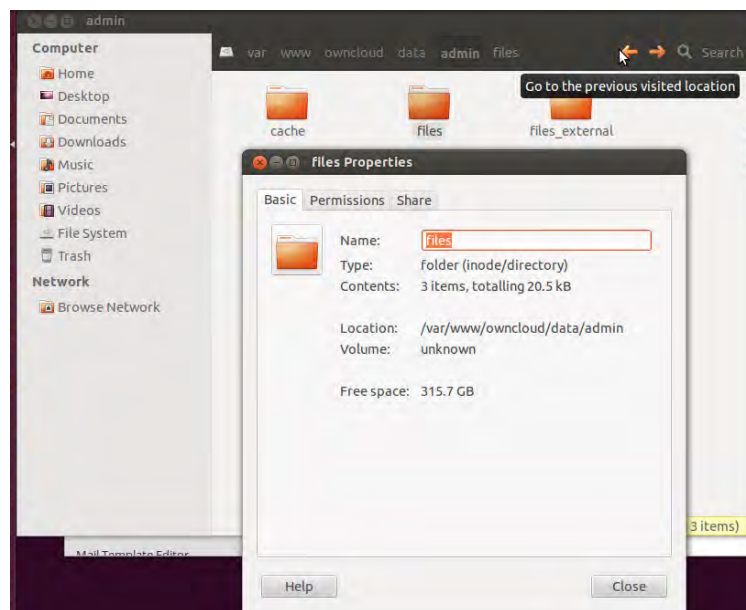


Gambar5. 15 Mounting ke Direktori Owncloud

Dapat dilihat seperti gambar 5.15 bahwa folder dari user admin telah berubah menjadi sebesar gabungan dari seluruh space drive yang ada.



Gambar 5.16 Akses Owncloud Melalui Web

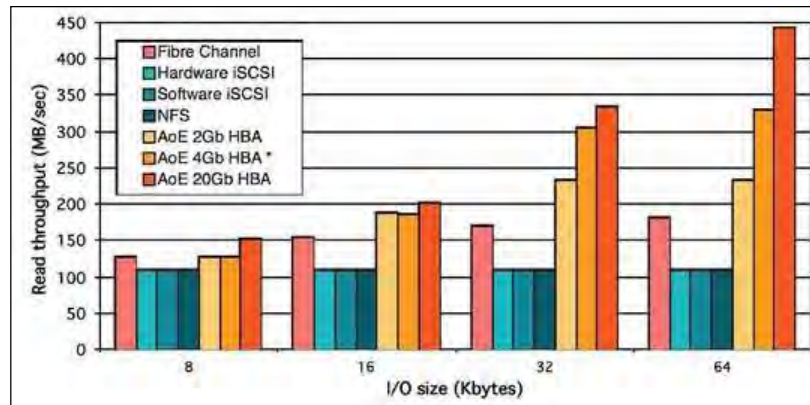


Gambar 5.17 Total Space Owncloud Storage Menggunakan LVM

Akhirnya dapat dilihat bahwa space owncloud pada server sudah sesuai dengan kapasitas storage total seperti yang sudah dikonfigurasi sebelumnya. Dimana storage total adalah gabungan dari beberapa storage yang lebih kecil. Namun dalam pelaksanaannya terdapat permasalahan pada saat penamnanan storage secara fisik pada server. Terdapat kesulitan menambahkan storage pada server karena keterbatasan slot media penyimpana di server. Unut itu perlu digunakan mekanisme ATA Over Ethernet dan Logical Volume Managemen agar media penyimpanan dapat ditambahkan berbasis IP.

5.3.2.3 ATA over Ethernet

ATA over Ethernet (AoE) merupakan protokol yang dikembangkan oleh *Brantley Coile Company*. Protokol ini memungkinkan sebuah *harddisk* diakses melalui kabel *ethernet* tanpa diperlukan *hardware* khusus. *ATA header command* yang berisi perintah seperti *read*, *write*, *status* dienkapsulasi dalam *ethernet frame* yang kemudian dikirimkan melalui jaringan *ethernet*. AoE berjalan pada *layer 2 ethernet* dan tidak menggunakan *internet protocol* (IP) sehingga AoE tidak dapat diakses melalui internet atau jaringan yang menggunakan IP. AoE bersifat *direct-mount* dari jaringan lokal *ethernet* dan hanya menggunakan *MAC Address* untuk pengalamatannya. Karena sifat protokolnya yang sederhana, AoE tergolong cepat dalam menangani proses *input/output*. (Huth,2011) Pada tahun 2009 Coraid merilis studi performa AoE yang menunjukkan, untuk ukuran *32Kbyteinput/output*, *throughput* yang dihasilkan AoE mencapai *234Mbyte* dibandingkan dengan protokol *Fibre Channel* yang menghasilkan *throughput* *172Mbyte* (Coile.B,2009).



Gambar 5.18. Hasil dari Coraid's Performance Study on AoE. Sumber : storagenewsletter.com.

AoE memungkinkan untuk membangun *Storage Area Network* dengan biaya rendah dan minim penggunaan *hardware* tambahan. Namun dibalik protokol AoE yang sederhana terdapat beberapa kelebihan dan kekurangan yang dimilikinya.

AoE memiliki beberapa kelebihan diantaranya :

- AoE bekerja pada *layer 2 OSI model*, sehingga protokol ini sangat sederhana, *low-latency*, dan mudah untuk diimplementasikan. AoE tidak menggunakan protokol rumit seperti pada *Fibre Channel* dan *iSCSI*
- Implementasinya yang tidak memakan biaya banyak dan tidak memerlukan *hardware* khusus.
- Tingkat keamanan yang lebih baik karena AoE menggunakan *MAC Address* sebagai sistem pengalamatan, sehingga menyebabkan protokol AoE *non-routable* jika dibandingkan protokol lain yang menggunakan IP.
- Protokol yang ringan dan cepat.

Disamping kelebihan yang telah dipaparkan, AoE bukanlah protokol yang sempurna dan masih memiliki beberapa kekurangan, diantaranya

- Sifat AoE yang *non-routable* memang menguntungkan dari sisi *security* namun membatasi skalabilitas jaringan AoE yang akan dibangun.
- Tingkat kesalahan yang tinggi ketika menggunakan media penyimpanan dengan sistem RAID.
- Tidak adanya *packet-loss detection* dan mekanisme *recovery*.

5.3.2.4 Logical Volume Management

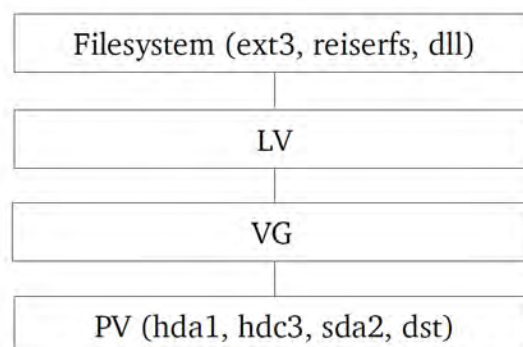
Logical Volume Management (LVM) adalah sebuah manajemen penyimpanan pada sistem operasi GNU/Linux yang menyediakan fleksibilitas dalam membuat dan mengubah partisi dalam sebuah disk. LVM menambahkan lapisan antara antarmuka *I/O kernel* dengan perangkat fisik untuk mendapatkan *logicalview* dari penyimpanan. Dengan adanya lapisan inilah kita dapat melakukan perubahan partisi baik itu *resize*, penghapusan, dan lain-lain secara *live* pada sistem baik untuk satu atau lebih media (Hasenstein,2001).

Kemudahan tersebut akan membuat administrator maupun pengguna lebih nyaman dalam melakukan manajemen dan alokasi ruang penyimpanan hingga penamaan *volume*. Selain itu kita dapat melakukan snapshot dan membuat *blockdevice* salinan sebuah partisi atau lebih untuk mencatat statusnya dengan tujuan backup.

Beberapa istilah dalam ruang lingkup LVM diantaranya:

- *Physical Volume (PV)*, adalah *physical disk* atau media penyimpan secara fisik baik itu berupa partisi secara konvensional misal: *hda1, hda3, hdc5, sda3* maupun RAID
- *Volume Group (VG)*, adalah sebuah *volume* yang dibuat dari satu atau lebih PV dalam media penyimpanan dan juga sebagai manajer bagi PV dan LV.
- *Logical Volume (LV)*, adalah partisi secara *logical* yang dibuat di atas VG dan pada LV inilah *filesystem* akan diletakkan.
- *Physical Extents (PE)*, adalah bagian atau potongan yang membentuk sebuah PV, dengan PE ini dapat ditentukan maksimal ukuran sebuah VG.
- *Logical Extents (LE)*, adalah bagian atau potongan yang membentuk LV, ukurannya sama dengan PE.

Berikut merupakan gambaran sederhana tentang struktur LVM.

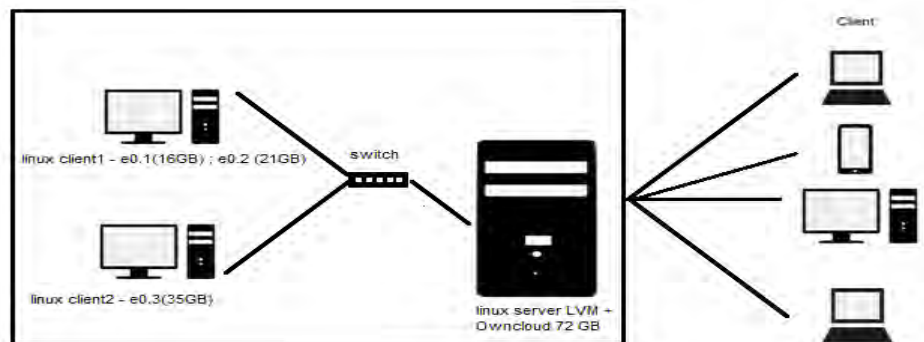


Gambar 5.19. Struktur LVM

LVM merupakan sistem yang sangat cocok jika digabungkan dengan AoE karena pada AoE akan menuntut manajemen *storage* untuk penanganan partisi yang banyak dari client.

5.3.2.5 SKEMA SISTEM

Pada sistem *Low-Cost Storage Area Network* yang dibangun akan menggunakan 1 *server* Ubuntu dan 2 *client* Ubuntu. Masing-masing *client* akan menjadi *initiator* dan server akan menjadi *target*. Masing-masing *client* akan menginisialisasikan partisi kedalam jaringan AoE dengan nama e0.x Berikut skema sistem *Low-Cost Storage Area Network*.



Gambar 5.20. Skema Sistem

Client1 akan menginisialisasikan 2 partisi yakni e0.1 dengan kapasitas 16GB dan e0.2 dengan kapasitas 21GB sedangkan Client2 menginisialisasikan 1 partisi yakni e0.3 dengan kapasitas 35GB. Setelah partisi AoE berhasil di-mount pada *server*, maka masing-masing partisi AoE akan disatukan dengan LVM sehingga *server* akan memiliki *space* tambahan dengan ukuran 72GB. Partisi LVM ini nantinya akan dimounting kepada suatu folder yang akan digunakan sebagai *data folder* pada owncloud.

5.3.2.6 IMPLEMENTASI AOE+LVM

Dalam pengimplementasian sistem Ubuntu 14.04 akan dipakai di sisi *server* dan Ubuntu 12.04 disisi *client*, sebuah *switch* diperlukan untuk mengkoneksikan jaringan. Diperlukan beberapa instalasi dan konfigurasi di masing-masing komputer baik itu *server* maupun *client* dengan melalui beberapa langkah sebagai berikut :

- a. Pastikan kernel Ubuntu sudah didukung oleh AoE. Ketikkan perintah `grep ATA_OVER /boot/config-`uname -r`` untuk mengeceknya. Jika kernel sudah didukung AoE maka akan dihasilkan output sebagai berikut:

```
root@aoclient-desktop:~# grep ATA_OVER /boot/config-`uname -r`
CONFIG_ATA_OVER_ETH=m
```

Gambar 5.22. Dukungan AoE Pada Server

- b. Pastikan *server* dan *client* sudah terkoneksi pada jaringan.
- c. Dilanjutkan dengan memuat modul dari AoE dengan mengetikkan perintah sebagai berikut dan pastikan module sudah dimuat.

```
root@aoeclient-desktop:~# sudo modprobe aoe
root@aoeclient-desktop:~# lsmod | grep aoe
aoe                49728  0
```

Gambar 5.23. Memuat Module AoE

- d. Untuk memuat modul secara otomatis setiap kali *reboot*, kita perlu menambahkan modul AoE pada file **/etc/modules** seperti gambar berikut.

```
modules ✖
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

loop
lp
rtc
aoe
```

Gambar 5.24. Memuat Modul Secara Otomatis

Setelah modul AoE sudah dimuat pada masing-masing *client* dan *server*, kemudian kita lanjutkan dengan menginisiasikan partisi pada *client* untuk dimuat pada jaringan AoE dengan beberapa langkah sebagai berikut :

- e. Install modul *vblade* pada *client* dengan mengetikkan perintah **apt-get install vblade**.
- f. Kemudian muat partisi yang akan diinisialisasikan ke *server* dengan perintah berikut.

```
root@aoeclient-desktop:~# vblade 0 2 eth0 /dev/sda4
pid 2658: e0.2, 50781465 sectors 0_RDWR
```

Gambar 5.25. Memuat Partisi /dev/sda4

Variabel 0 merupakan *shellf number (major)* dan 2 merupakan *slot number (minor)* variabel tersebut digunakan untuk mengidentifikasi partisi AoE dan dapat dikondisikan untuk masing-masing partisi namun tidak boleh menggunakan kombinasi variabel yang sama. *Range* yang dimiliki variabel *major* yakni 0 – 65535 sedangkan *range minor* dari 0 – 255. Bagian **eth0** menunjukkan perangkat *ethernet* yang digunakan dan **/dev/sda4** menunjukkan partisi yang dimuat. Ketika sudah menunjukkan output *pid* maka partisi sudah berhasil dimuat.

- g. Untuk memuat partisi secara otomatis setiap kali *reboot* perintah *vblade* dapat ditambahkan pada file **/etc/rc.local** seperti pada gambar dibawah.

```
# By default this script does nothing.

vblade 0 1 eth0 /dev/sda3
vblade 0 2 eth0 /dev/sda4

exit 0
```

Gambar 5.26. Konfigurasi Vblade

Tambahkan *command vblade* sebelum *exit 0*. Beberapa partisi dapat dimuat secara bersamaan seperti pada gambar yang memuat *sda3* dan *sda4*.

Setelah *client* berhasil menginisialisasi masing-masing partisi kemudian cek masing-masing partisi apakah sudah dimuat pada *server* untuk nantinya akan dilakukan *formatting* dan *setup LVM*. Lakukan langkah-langkah berikut pada komputer *server*.

- a. Install *aoetools* dengan mengetikkan perintah `apt-get install aoetools`.
- b. Cek partisi AoE dengan mengetikkan perintah **aoe-discover** kemudian **aoe-stat** maka akan tampil keluaran sebagai berikut.

```
root@ilkom-AoE:~# aoe-stat
e0.1      16.680GB      eth0 1024 up
e0.2      26.000GB      eth0 1024 up
e0.3      35.468GB      eth0 1024 up
```

Gambar 5.27. AoE Status

Partisi yang berhasil dimuat ke *server* diidentifikasi dengan partisi *e0.1*, *e0.2* dan *e0.3* yang masing-masing dimuat melalui *eth0* dengan status *up*.

- c. Kemudian format partisi AoE dengan menggunakan *file system ext4*
- d. Setelah melakukan *format file system*, lihat detail partisi dengan `fdisk -l` maka akan tampil keluaran sebagai berikut.

```
Disk /dev/etherd/e0.1: 16.7 GB, 16680867840 bytes
255 heads, 63 sectors/track, 2028 cylinders, total 32579820 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x88cf3ba9

    Device Boot      Start         End      Blocks   Id  System
 /dev/etherd/e0.1p1    2048        32579819    16288886   83   Linux

Disk /dev/etherd/e0.2: 26.0 GB, 26000110080 bytes
255 heads, 63 sectors/track, 3161 cylinders, total 50781465 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd995dc95

    Device Boot      Start         End      Blocks   Id  System
 /dev/etherd/e0.2p2    2048        50781464    25389708+   83   Linux

Disk /dev/etherd/e0.3: 35.5 GB, 35468083200 bytes
255 heads, 63 sectors/track, 4312 cylinders, total 69273600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000d770f

    Device Boot      Start         End      Blocks   Id  System
 /dev/etherd/e0.3p1    2048        69273599    34635776   83   Linux
```

Gambar 5.28. Logical Partition Pada Server

Partisi *e0.1*, *e0.2* dan *e0.3* akan menjadi *device e0.1p1*, *e0.2p2* dan *e0.3p1* yang berada pada direktori */dev/etherd/*. Partisi tersebut sudah siap untuk digabungkan dengan LVM.

5.3.3 Uji coba Pada LVM + AoE

Setelah melakukan scenario percobaan dengan *ATA over Ethernet* didapat bahwa partisi dari *client* berhasil dimuat ke server, yang kemudian pada server partisi-partisi tersebut akan dibuat menjadi *logical volume* untuk memudahkan dalam hal manajemen. LVM dapat dibuat dengan menginstall aplikasi *lvm2* pada Ubuntu dan membuat *physical volume* dari partisi AoE dengan perintah **pvcreate**.

```
root@ilkom-AoE:~# pvs
PV                VG      Fmt Attr PSize PFree
/dev/etherd/e0.1p1  lvm2 a-- 15,53g 15,53g
/dev/etherd/e0.2p2  lvm2 a-- 24,21g 24,21g
/dev/etherd/e0.3p1  lvm2 a-- 33,03g 33,03g
root@ilkom-AoE:~#
```

Gambar 5.29. *Physical Volume* dari PartisiAoE

Kemudian *physical volume* disatukan menjadi sebuah *volume group* dengan perintah **vgcreate**

```
root@ilkom-AoE:~# vgs
VG      #PV #LV #SN Attr   VSize VFree
vgaoe   3   0   0 wz--n- 72,77g 72,77g
root@ilkom-AoE:~#
```

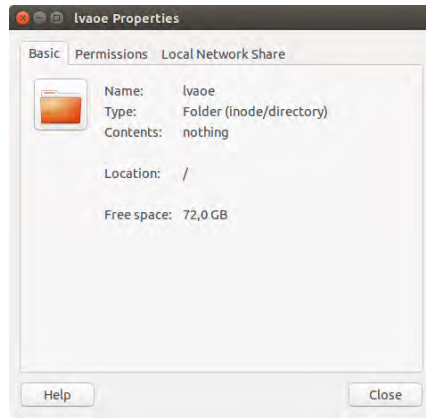
Gambar 5.30. *Volume Group* dari PartisiAoE

Dari *volume group* yang tersedia maka bisa dibuat beberapa *logical volume* dengan perintah **lvcreate**.

```
root@ilkom-AoE:~# lvcreate -n lvaoe -L 72G vgaoe
Logical volume "lvaoe" created
root@ilkom-AoE:~# lvs
LV      VG      Attr      LSize Pool Origin Data% Move Log Copy% Convert
lvaoe   vgaoe  -wi-a---- 72,00g
root@ilkom-AoE:~#
```

Gambar 5.31. *Logical Volume* dari PartisiAoE

Kemudian *logical volume* tersebut dapat di-mount kedalam *file system* dengan menambahkan direktori pada *root*. Pada system yang dibangun ,direktori *"/lvaoe"* akan dibuat dan dipakai sebagai *mount point* dari system. Setelah direktori *"/lvaoe"* dibuat, tambahkan *script* pada file *"/etc/fstab"* untuk melakukan *automatic mounting* setiap kali *reboot*. Edit file *"/etc/fstab"* dengan menambahkan *"/dev/vgaoe/lvaoe"* pada kolom <file system>, *"/lvaoe"* pada kolom <mount point>, *"ext4"* pada kolom <type>, *"defaults"* pada kolom <options>, variable *0* pada kolom <dump> dan <pass>. Setelah mengedit file *"/etc/fstab"* , ketikkan perintah *"mount/lvaoe"* maka server sudah bias memakai direktori yang telah dibuat dan menambah kapasitas penyimpanan server sebanyak 72GB.

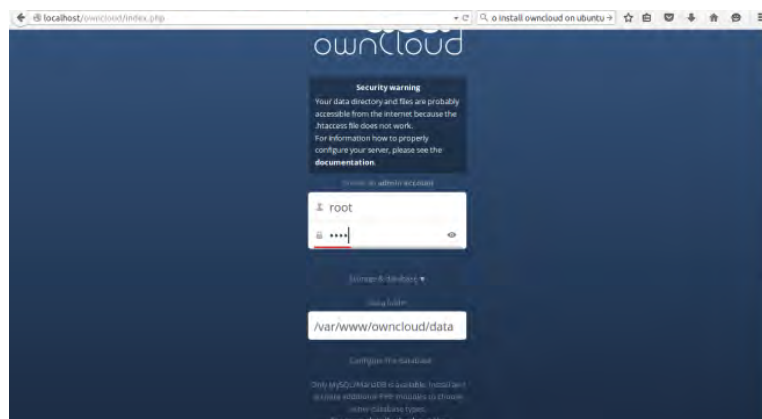


Gambar 5.32. Hasil akhir dari Sistem AoE-LVM.

Setelah mengimplementasikan system, didapat bahwa partisi AoE berhasil digabungkan dengan LVM dan sudah mampu menambah ruang penyimpanan pada server sebanyak 72GB. Hal yang perlu diingat pada system AoE-LVM ini adalah bahwa partisi yang diambil dari *client* akan menyatu dengan *file system* pada server, sehingga ketika salah satu *client* mati atau terputus darikoneksi jaringan, maka itu akan ada kemungkinan untuk membuat data pada *file system* yang telah dibuat menjadi tidak bisa dibaca atau rusak. Selain itu, ketika proses *booting server* akan menunggu sampai semua *client* AoE terkoneksi untuk *mount file system* yang telah dibuat dan jika salah satu *client* terputus saat proses *booting* maka *server* akan terus menunggu sampai *client* terkoneksi atau partisi dari AoE telah di-*skip mounting* dari *file system* ketika *booting*.

5.3.4 Uji Coba Pada Owncloud

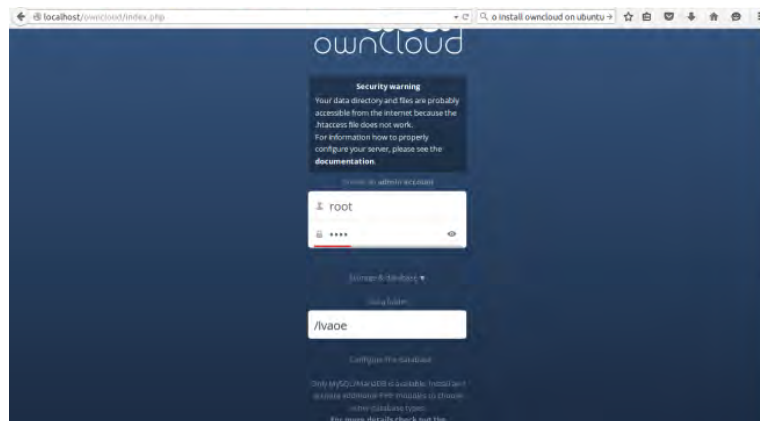
Setelah melakukan konfigurasi AoE dan LVM pada server, maka tahap selanjutnya yaitu melakukan konfigurasi pada owncloud. Pastikan owncloud sudah terinstall dengan benar seperti gambar dibawah ini :



Gambar 5.33. Tampilan Awal Konfigurasi Owncloud

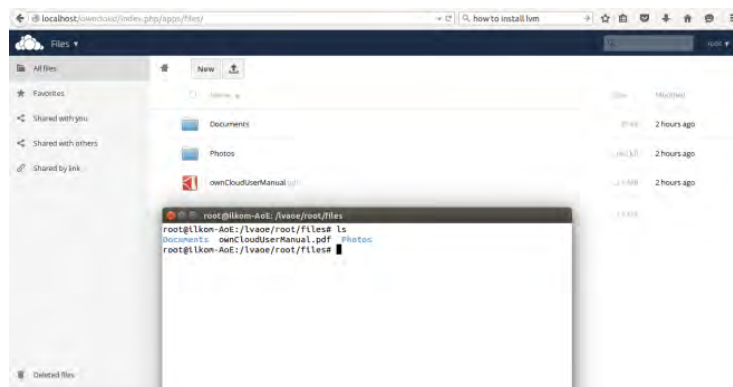
Terlihat pada kolom *data folder* , secara default owncloud akan mengarah pada `/var/www/owncloud/data`, ubah menjadi folder tempat AoE folder yang telah di konfigurasi

tadi (dalam penelitian ini berada pada /lvaoe) dan ubah kepemilikan folder /lvaoe menjadi www-data agar folder tersebut dapat diakses oleh owncloud seperti gambar dibawah ini :



Gambar 5.34. Konfigurasi Data Folder

Setelah melakukan konfigurasi berhasil maka akan tampak pada folder /lvaoe akan berisi data – data client pada owncloud :



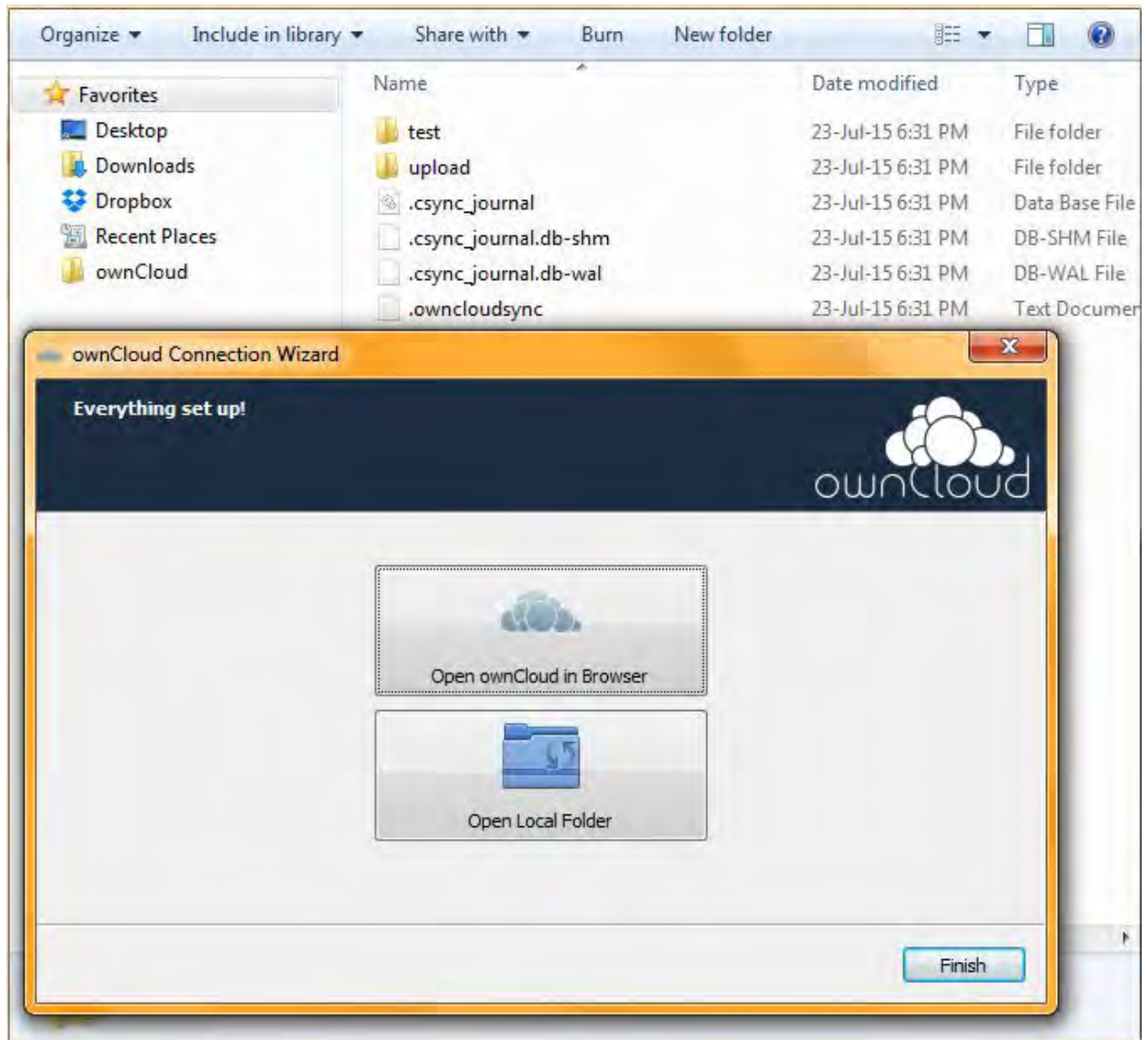
Gambar 5.35. Konfigurasi Data Folder Berhasil

Dari gambar 5.35 terlihat konfigurasi owncloud sudah berhasil digabungkan dengan folder AoE sehingga kapasitas penyimpanan pada owncloud bertambah besar.

5.3.5 Ujicba User Owncloud

untuk memastikan dan menguji apakah sistem sudah mampu melakukan tugas seperti yang telah direncanakan sebelumnya, maka perlu dijalankan beberapa scenario ujicoba. Adapun ujicoba yang akan dilakukan adalah dengan membuat sebuah account user dan user tersebut akan diberikan akses login ke owncloud untuk selanjutnya diperiksa kapasitas penyimpanan cloud apakah sudah sesuai dengan yang di konfigurasi sebelumnya.

Untuk itu akan dijalankan akses melalui owncloud client yang ada pada system operasi windows. Kita melakukan konfigurasi pada client dimana alamat server dari owncloud server yaitu merupakan alamat IP local dimana tempat server terinstall, lalu pilih open local folder maka akan tampil folder yang ada pada owncloud server sama dengan yang ada pada owncloud client seperti gambar 5.18



Gambar 5.19 Owncloud Client

Demikian konfigurasi LVM dengan owncloud yang telah berhasil pada sisi client tidak akan mengetahui data yang disimpan terdapat di harddrive mana pada sever karena LVM yang akan mengatur penyimpanan data client yang akan ditempatkan pada ke empat fisik harddisk yang tersedia.

BAB 6

KESIMPULAN DAN SARAN

Rencana yang akan dilaksanakan pada tahap berikutnya adalah :

1. Sistem validasi karya ilmiah yang telah dibangun telah berfungsi dengan normal dan memenuhi kebutuhan validasi karya ilmiah di lingkungan Universitas Udayana. Hal ini dapat dilihat dari hasil analisa pengujian fungsional system yang berjalan sesuai dengan kebutuhan yang ada.
2. Penggabungan AoE sebagai storage pada Owncloud dengan LVM adalah solusi yang dapat dilakukan untuk menambah storage pada owncloud, dengan hal ini maka client dapat melakukan upload file dalam skala yang lebih besar, sehingga keterbatasan media penyimpanan pada owncloud dapat teratasi.

DAFTAR PUSTAKA

BPMU Universitas Udayana, 2012, "*Standar Operasional Prosedur Validasi Karya Ilmiah*", Universitas Udayana

CAT, 2010. "Cloud Computing Basics".

http://www.south.cattellecom.com/Technologies/CloudComputing/0071626948_chap01.pdf. 27 April 2013.

Coile, B. S. Hopkins 2009. The ATA over Ethernet Protocol. Coraid, Inc. 565 Research Drive.

Direktorat Jenderal Pendidikan Tinggi, 2010, "*Pedoman beban kerja dosen dan evaluasi pelaksanaan tridharma perguruan tinggi*", Departemen Pendidikan Nasional

Firdaus, H. B. 2008. *Deteksi Kesamaan Dokumen Menggunakan Algoritma Rabin-Karp*. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung (ITB). Bandung.

Gipp dan Meuschke. 2011. *Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence*. Proceedings of the 11th ACM Symposium on Document Engineering (DocEng2011).

Hasenstein, Michael, 2001. The Logical Volume Management. SuSE Inc.

HowtoForge.com 2012. Using ATA Over Ethernet (AoE) On Ubuntu 12.10 (Initiator and Target) . <URL:<https://www.howtoforge.com/using-ata-over-ethernet-aoe-on-ubuntu-12.10-initiator-and-target>>.

Huth, A., Cebula, J. 2011. "The basics of cloud computing". Carnegie Mellon University.

Salmusih dan Sunyoto, Andi. 2013. *Implementasi Algoritma Rabin-Karp untuk Pendeteksi Dokumen Teks Menggunakan Konsep Similarity*. Teknik Informatika, STIMIK AMIKOM Yogyakarta. Yogyakarta. ISSN: 1907 – 5022.

Michael A. Covington, Ph.D. 2008. An Overview of CORAID Technology and ATA-over-Ethernet (AoE). Georgia : University of Georgia.

Kleijn ,Gerben, February 2013. “ATA Over Ethernet (AoE)”.NTW 412.

StorageNewsLetter.com 2009. ATA-Over-Ethernet Has Better Throughput Than FC
<URL<http://www.storagenewsletter.com/rubriques/systems-raid-nas-san/coraid-ata-over-ethernet-better-throughput-fc/>>.

Surahman, A.M. 2013. *Perancangan Sistem Penentuan Similarity Kode Program pada Bahasa C dan Pascal dengan Menggunakan Algoritma Rabin-Karp*. Jurnal Sistem dan Teknologi Informasi (JustIN), 1(1).

LAMPIRAN

1. Justifikasi Anggaran Penelitian

Rincian Penggunaan

1. HONOR OUTPUT KEGIATAN				
Item Honor	Volume	Satuan	Honor/Jam (Rp)	Total (Rp)
1. Honor 2 orang anggota Peneliti mahasiswa Bulan mei	2.00	OB	400.000	800.000
2. Honor 2 orang anggota Peneliti Bulan mei	2.00	OB	500.000	1.000.000
3. Honor Ketua Peneliti Bulan mei	1.00	OB	600.000	600.000
4. Honor Ketua Peneliti Bulan Juni	1.00	OB	600.000	600.000
5. Honor 2 orang anggota Peneliti Bulan Juni	2.00	OB	500.000	1.000.000
6. Honor Ketua Peneliti Bulan Juli	1.00	OB	600.000	600.000
7. Honor 2 orang anggota Peneliti Bulan Juli	2.00	OB	500.000	1.000.000
8. Honor 2 orang anggota Peneliti mahasiswa Juli	2.00	OB	400.000	800.000
9. Honor 2 orang Anggota Penelitain mahasiswa Juni	2.00	OB	400.000	800.000
10. Honor Ketua Peneliti Bulan Agustus	1.00	OB	600.000	600.000
11. Honor 2 orang anggota Peneliti Bulan Agustus	2.00	OB	500.000	1.000.000
12. Honor 2 orang Anggota Penelitain mahasiswa Agustus	2.00	OB	400.000	800.000
13. Narasumber pembuatan sistem	2.00	OB	600.000	1.200.000
Sub Total (Rp)				10.800.000,00

26. Konsumsi Progress Report 3 Juli 2015	1.00	Paket	100.000	100.000
27. Konsumsi Progress Report 19 Juni 2015	1.00	Oaket	100.000	100.000
28. Toner Printer Samsung ML2240 12 Juni 2015	1.00	Buah	800.000	800.000
29. Konsumsi Progress Report 10 Juli 2015	1.00	Paket	100.000	100.000
30. konsumsi progress report	1.00	Paket	100.000	100.000
31. WD Purple 6TB - cloud storage	2.00	Buah	4.320.000	8.640.000
32. harddisk WD My Passport Ultra New 1TB USB 3.0	3.00	Buah	1.360.000	4.080.000
33. Kertas HVS	2.00	RIM	34.000	68.000
34. Toner Samsung ML2240	1.00	buah	1.000.000	1.000.000
35. konsumsi progress report	1.00	Paket	100.000	100.000
36. konsumsi progress report	1.00	Paket	100.000	100.000
37. konsumsi progress report	1.00	paket	100.000	100.000
38. lokakarya dan diskusi	1.00	paket	2.000.000	2.000.000
39. analisis data senastek	1.00	paket	500.000	500.000
40. konsumsi progress report	1.00	paket	200.000	200.000
41. konsumsi progress report	1.00	paket	200.000	200.000
42. Paket Internet Anggota mahasiswa 3 bulan	6.00	OB	150.000	900.000
43. konsumsi progress report	1.00	paket	200.000	200.000
44. laporan dan luaran	10.00	buah	50.000	500.000
45. biaya publikasi di SENASTEK 2015	1.00	paket	1.000.000	1.000.000
46. publikasi seminar internasional	1.00	paket	4.000.000	4.000.000
47. Konsumsi FGD	1.00	Paket	300.000	300.000
48. Pajak Honor	1.00	Paket	1.080.000	1.080.000
49. Pajak ATK dan Bahan	1.00	paket	1.779.119	1.779.119
50. pajak konsumsi	1.00	Paket	316.250	316.250
Sub Total (Rp)				34.013.969,00

Item Barang	Volume	Satuan	Harga Satuan (Rp)	Total (Rp)
1. Biaya Konsultasi	1.00	Paket	200.000	200.000
2. akomodasi	6.00	OH	800.000	4.800.000
3. Lumsum	6.00	OH	400.000	2.400.000
Sub Total (Rp)				7.400.000,00
4. BELANJA PERJALANAN LAINNYA				
Item Perjalanan	Volume	Satuan	Biaya Satuan (Rp)	Total (Rp)
1. Transportasi diskusi	5.00	Buah	100.000	500.000
2. Perjalanan Ke tempat seminae	3.00	Orang	1.600.000	4.800.000
Sub Total (Rp)				5.300.000,00
Total Pengeluaran Dalam Satu Tahun (Rp)				57.513.969,00



Mengetahui,
 Ketua Lembaga Penelitian dan Pengabdian Kepada Masyarakat Universitas Udayana
 (Drs. I Nyoman Gde Antara, M.Eng)
 NIP/NIK 196408071992031002

Jimbaran, 10 - 11 - 2015
 Ketua,

(AGUS MULLANTARA S.Kom.,M.Kom.)
 NIP/NIK 198006162005011001