

# MEMBANGUN PRIVATE CLOUD COMPUTING MENGGUNAKAN SISTEM OPERASI PROXMOX PADA RUMAH SAKIT SAMARINDA MEDIKA CITRA

Ahmad Rofiq Hakim <sup>1)</sup>, Asep Nurhuda <sup>2)</sup>, Nurul Hadi <sup>3)</sup>

<sup>1,2,3</sup> Teknik Informatika, STMIK Widya Cipta Dharma

<sup>1,2,3</sup> Jl. Prof. M. Yamin No.25, Samarinda, 75123

E-mail: rofiq\_93@yahoo.com<sup>1)</sup>, asep.noor@gmail.com<sup>2)</sup>, nurulhadi411@gmail.com<sup>3)</sup>

## ABSTRAK

Penelitian dilakukan untuk dapat membuat sebuah *virtual server* berbasis *Open Source* yang nantinya jika penelitian ini berhasil bisa membantu Rumah Sakit Samarinda Medika Citra dalam memenuhi kebutuhan *Server* yang dapat di sesuaikan dengan keperluan Sistem Informasi Rumah Sakit.

Dalam penelitian ini metode pengembangan sistem yang di gunakan yaitu PPDIOO (*Prepare, Plan, Design, Implement, Operate and Optimize*) dengan sistem operasi yang di gunakan adalah *Proxmox* yang merupakan distro linux debian (x86\_64) yang bersifat *Open Source*.

Adapun hasil akhir dari penelitian ini yakni berupa *virtual-virtual server* yang dapat di gunakan untuk beberapa layanan aplikasi di antaranya yaitu *Owncloud, Inacbgs, EyeOS* dan *Proxy Server* yang dapat menghemat anggaran pembelian komputer *server* Departement Teknologi Informasi Rumah Sakit Samarinda Medika Citra.

Kata Kunci: *Virtual Server, Cloud Computing, Proxy, Owncloud, EyeOS, InaCbg*

## 1. PENDAHULUAN

*Cloud Computing* adalah sebuah teknologi sistem komputerisasi berbasis jaringan *internet* maupun *intranet*, dimana suatu sumber daya *hardware* dan *software* disediakan sebagai sebuah layanan untuk di gunakan oleh komputer lain yang membutuhkan. *Cloud Computing* bisa di anggap sebagai perluasan dari virtualisasi. Aplikasi atau sistem yang di gunakan bisa di tempatkan di dalam jaringan.

*Proxmox hypervisor* adalah salah satu yang terbaik di dunia dalam melakukan virtualisasi karena kestabilan kinerja, kemudahan dalam penggunaannya dan juga merupakan sistem operasi yang bersifat *Open Source*. *Proxmox* menyertakan dua teknologi virtualisasi sekaligus yaitu OpenVZ dan KVM (*Kernel-based Virtual Machine*) yang memungkinkan dapat membuat *virtual machine* dengan berbagai sistem operasi seperti windows, linux, mikrotik dan sistem operasi lainnya. Sistem Operasi *Proxmox* yang merupakan distro linux debian (x86\_64) memang di khususkan sebagai distro virtualisasi dengan mesin 64 bit dan tidak dapat di gunakan di mesin 32 bit.

Dokter Spesialis dengan Visi yang sama yaitu membangun sebuah rumah sakit unggulan di kota samarinda. Dari ide tiga orang Dokter, tujuh orang pengusaha dan satu bidan. Dengan adanya pertemuan-pertemuan dan diskusi serta kesepakatan bersama maka di mulailah pembangunan Rumah Sakit Medika Citra yang kemudian melakukan pelayanan kesehatan melalui *soft opening* pada tanggal 12 Juli 2013 di jadikan sebagai

hari jadi Rumah Sakit Samarinda Medika Citra.

Masalah yang ada pada Rumah Sakit Umum SMC pada saat ini yaitu besarnya biaya yang harus di keluarkan untuk membeli *software* maupun *hardware*. Komputer *Server* sistem informasi hanya ada satu yakni *Server* sistem informasi rumah sakit (SIRS) dan *Server* replikasi sedangkan untuk file *Server* tidak maksimal serta penambahan *Server* inacbg belum ada. Infrastruktur jaringan yang memiliki *IP* dinamis menjadi salah satu masalah dalam akses *Server* lokal secara *public* via internet.

Berdasarkan latar belakang tersebut di atas, maka dirasa perlu untuk membuat sebuah penelitian dengan judul “**Membangun Private Cloud Computing Menggunakan Sistem Operasi Proxmox Pada Rumah Sakit Samarinda Medika Citra**”. Sehingga harapannya dapat meningkatkan efisiensi dalam menghemat anggaran belanja *proprietary software* dan *hardware* dalam mengoptimalkan dan mengintegrasikan pemanfaatan teknologi informasi dan komputer terbaru.

## 2. RUANG LINGKUP PENELITIAN

Dalam penelitian ini permasalahan mencakup:

- 1 Sistem Operasi *Cloud Computing* menggunakan *Proxmox* VE 3. 4.
- 2 Penelitian menggunakan metode PPDIOO (*Prepare, Plan, Design, Implement, Operation, Optimize*).

- 3 Model *Deployment Cloud Computing* yang di bangun adalah *Private Cloud Computing*.
- 4 Jenis layanan yang di gunakan adalah Saas (*Software as a Service*).
- 5 Jaringan *Internet* Menggunakan IP *public* dinamis dengan memanfaatkan fitur *IP Cloud* dari *Mikrotik OS*.
- 6 Studi kasus aplikasi yang berjalan di *Server Cloud* adalah *EyeOS*, *Own Cloud*, *Proxy Server* dan *INA CBGs*

### 3. BAHAN DAN METODE

#### 3.1 Cloud Computing

Menurut Purbo (2012), *Cloud Computing* adalah sebuah model 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 umumnya mirip dengan modem layanan *Public*. Ketersediannya on-demand sesuai kebutuhan, mudah dikontrol, dinamik dan skalabilitas yang hampir tanpa limit adalah beberapa atribut penting dari *Cloud Computing*.

#### 3.2 Proxmox VE

Menurut *Web Resmi Proxmox*, ([proxmox.com](http://proxmox.com) diakses pada tanggal 30 Maret 2015 dan diterjemahkan oleh google translate) *Proxmox VE* adalah solusi virtualisasi manajemen yang lengkap sumber terbuka untuk *Server*. Hal ini didasarkan pada virtualisasi KVM dan virtualisasi berbasis kontainer dan mengelola mesin *virtual*, penyimpanan, jaringan *virtual*, dan *HA Clustering*.

Fitur kelas *enterprise* dan antarmuka web yang intuitif dirancang untuk membantu Anda meningkatkan penggunaan sumber daya yang ada dan mengurangi biaya *hardware* dan pengadministrasian waktu didalam berbisnis serta digunakan di rumah. Bahkan Anda dapat dengan mudah dalam membuat virtualisasi.

#### 3.3 Proxy Server

*Proxy Server* menurut komputer (2008) adalah komponen yang terletak antara komputer *Client* dengan *internet*. Ketika komputer-komputer *Client* hendak mengakses *internet*, mereka tidak langsung terkoneksi ke *WWW* dan mengirimkan *HTTP request*, namun konek dahulu ke *proxy Server* lokal. *Proxy* kemudian mem-forward *request* tersebut ke *WWW*, mengambil hasilnya dan kembali mengembalikan lagi ke *Client*. *Proxy* bertindak sebagai *gateway* bagi setiap komputer *Client* -nya. Gambar berikut menggambarkan posisi dan fungsi dari *proxy Server*, diantara pengguna dan penyedia layanan.

#### 3.4 Owncloud

Menurut Afrianto (2013), *OwnCloud* merupakan salah satu perangkat lunak berbagi berkas gratis (lisensi *AGPL*) dan bebas disertai juga edisi bisnis dan edisi

perusahaan, menyediakan pengamanan yang baik, memiliki tata cara yang baik bagi pengguna aplikasi untuk membagi dan mengakses data yang secara terintegrasi dengan perangkat teknologi informasi yang tujuannya mengamankan, melacak, dan melaporkan penggunaan data.

#### 3.5 INACBG's

*INACBG's (Indonesian Case Base Groups)* adalah metode pembayaran pada program jaminan kesehatan masyarakat (*JAMKESMAS*) dengan menggunakan sistem *casemix* sesuai dengan Peraturan Presiden Nomor 12 Tahun 2013 tentang Jaminan Kesehatan sebagaimana telah diubah dengan Peraturan Presiden Nomor 111 Tahun 2013. Untuk tarif yang berlaku pada 1 Januari 2014, telah dilakukan penyesuaian dari tarif *INA-CBG Jamkesmas* dan telah ditetapkan dalam Peraturan Menteri Kesehatan Nomor 69 Tahun 2013 tentang Standar Tarif Pelayanan Kesehatan pada Fasilitas Kesehatan Tingkat Pertama dan Fasilitas Kesehatan Tingkat Lanjutan dalam penyelenggaraan Jaminan Kesehatan. Sistem *casemix* adalah pengelompokan *diagnosis* dan prosedur dengan mengacu pada ciri klinis yang mirip/sama dan biaya perawatan yang mirip/sama, pengelompokan dilakukan dengan menggunakan *groupier*.

#### 3.6 EyeOS

Menurut [linuxjournal.com](http://linuxjournal.com) (diakses pada 12 nopember 2015) *eyeOS* adalah sebuah sistem operasi *desktop* berbasis *web*. Meskipun orientasi penyebaran yang tidak biasa, dalam banyak hal, itu adalah sistem operasi *desktop* penuh, lengkap dengan manajemen *file* dan aplikasi lengkap yang beroperasi dalam *draggable*, jendela *resizable*. Serta gambaran tentang apa yang *eyeOS 1,9* cabang yang ditawarkan, saya akan membawa Anda melalui apa yang saya lakukan untuk menginstalnya pada sistem operasi jenis *Debian*. Secara keseluruhan, saya merasa bahwa itu adalah sistem yang dapat digunakan dan mungkin bahkan sekilas apa yang paling komputasi akan menjadi seperti di masa depan. Saya akan berkonsentrasi pada pengalaman saya mendirikan *eyeOS 1,9* daripada versi *2.x* yang diluncurkan awal tahun ini. Aku mencoba *eyeOS 2.x* pada beberapa set up yang berbeda, tapi aku terus berlari ke dalam masalah kinerja yang serius yang dilaporkan secara luas di forum *eyeOS* oleh orang lain yang telah mencoba. Seri *1.x* juga memiliki keuntungan dari yang telah dianggap stabil selama lebih dari dua tahun.

#### 3.7 Metode Pengembangan Sitem

Suatu sistem yang baik dirancang dengan menggunakan perencanaan dan mengikuti salah satu model proses perangkat lunak yang sesuai dengan sistem yang akan dibangun. Suatu model proses adalah juga dikenal sebagai sebagai suatu paradigma teknik pengembangan perangkat lunak. Model proses tersebut dipilih didasarkan dari kebutuhan sebuah sistem. Masing-masing model menunjuk tahap-tahap umum yang diperlukan, antara lain tahap definisi, tahap

pengembangan dan tahap pemeliharaan. Berikut adalah model pengembangan system yang digunakan:

### 1. PPDIOO (*Prepare, Plan, Design, Implement, Operate, Optimize*)

Menurut Sofana (2012), pada sebuah *network* yang dinamis, pengembangan *network* merupakan sebuah siklus yang berjalan terus menerus. Tidak pernah berhenti. Jika akan membuat sebuah *network* maka perencanaan merupakan hal yang sangat penting. Tanpa adanya perencanaan yang matang maka kita akan sulit mengantisipasi kebutuhan dimasa yang akan datang. Secara umum, *lifecycle* pengembangan *network* mengikuti pendekatan PPDIOO (*Prepare, Plan, Design, Implement, Operate, And Optimize*). Berikut ini penjelasannya.

#### 1) Prepare

Mencakup persiapan ditingkat organisasi, seperti menentukan strategi *network*, menentukan model bisnis yang sesuai dengan *network* yang akan dibangun.

#### 2) Plan

Merencanakan kebutuhan *network*, melakukan analisis, dan perencanaan waktu pengerjaan proyek

#### 3) Design

Membuat desain *network* secara detail.

#### 4) Implement

Melakukan implementasi berdasarkan rencana dan desain yang sudah dibuat.

#### 5) Operate

Pengoperasian *network* dan memonitor kondisi *network*

#### 6) Optimize

Melakukan optimasi dan mengatasi kelemahan yang ada.

### 3.8 Metode Pengujian Sistem

Menurut Pressman (2007) Pengujian perangkat lunak dalam buku rekayasa perangkat lunak pendekatan praktisi jilid dua adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari *spesifikasi, desain, dan pengkodean*.

Dalam tahap pengujian ini penulis menggunakan *black box*. Alasannya karena pengujian *black box* dapat mengetahui apakah perangkat lunak yang dibuat dapat berfungsi dengan benar dan telah sesuai dengan yang diharapkan. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika *intern* perangkat lunak. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibandingkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Dalam tahap pengujian terdapat metode-metode untuk melakukan

pengujian yaitu diantaranya:

#### 1. White Box

Metode yang menggunakan struktural kontrol desain prosedural. Jawaban yang dicari dari metode ini adalah memberikan jaminan bahwa semua jalur *program* pada modul paling tidak digunakan satu kali, menggunakan keputusan logis pada sistem dan eksekusi semua *loop* pada batasan mereka dan batasan operasional mereka.

#### 2. Black Box

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perencana perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu *program*.

#### 3. Metode Alpha Testing

Pengujian yang dilakukan oleh pengembang dari aplikasi / sistem yang akan dibangun. Dilakukan pada sisi pengembang oleh seorang pelanggan. Aplikasi digunakan pada *setting* yang natural dengan pengembang yang memandang melalui bahu pemakai dan merekam semua kesalahan dan masalah pemakaian.

#### 4. Metode Beta Testing

Pengujian yang dilakukan oleh pihak pengguna dari aplikasi yang akan dibangun. Dilakukan pada satu atau lebih pelanggan oleh pemakai aplikasi dalam lingkungan sebenarnya.

#### 5. Metode Structure Control

Pengujian mendeteksi bukan hanya kesalahan didalam kondisi *program*, tetapi juga kesalahan pada lain *program*.

### 4. RANCANGAN SISTEM

Sesuai dengan metode yang di gunakan tahapan rancangan sistem meliputi:

#### 1. Prepare

Analisa permasalahan yang muncul belum tersedianya *Server INACBG's* serta tidak maksimalnya penggunaan *File Server* yang di sebabkan kesulitan dalam mengakses *Server* tersebut oleh *client*. Pada tahap ini di lakukan identifikasi permasalahan dari penelitian ini yakni bagaimana membuat sistem *cloud computing* yang dapat menyediakan layanan SaaS (*Software As A Services*). Peneliti akan menggunakan empat layanan SaaS yaitu *Inacbg's, Proxy Server, Owncloud, dan EyeOS*

#### 2. Plan

Merencanakan kebutuhan sistem baik itu secara *hardware* maupun *Software* yang akan di buat dan di harapkan dapat memberikan gambaran seutuhnya terhadap kebutuhan yang ada.

**Tabel 1. Spesifikasi Hardware yang akan di gunakan**

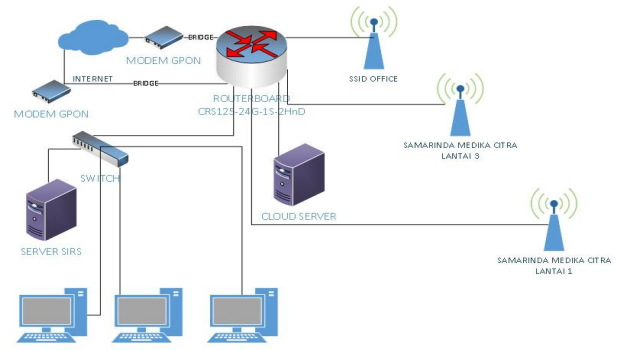
NO	Perangkat Keras	Spesifikasi
1	Processor	Xeon E5-2407 2.20Ghz
2	CPU 64 Bit	64 Bit
3	Virtualization	YES
4	Random Acces Memory	4 GB
5	Hardware Raid	RAID
6	Fast Hard Drives	15000 RPM 500GB
7	Network Card	1 Card support Gigabit

**Tabel 2. Perangkat jaringan**

NO	Perangkat Jaringan	Spesifikasi	Unit
1	Routerboard Mikrotik	Cloud Core Switch (CRS) Gps	1
2	Switch/Hub	Tp-link 32 Port 1Gbps	1
3	Access Point	Unifi AP	1
		TP-Link WR841N	1
		D-Link DIR-600	1
4	Kabel STP	Belden STP Cat 6	1
5	Modem GPON	Alcatel Lucent i- 240w-a	2

3. *Design*

Pada tahap desain topologi jaringan yang saat ini berjalan dan yang akan di gunakan di dalam penelitian ini Rumah Sakit Samarinda Medika Citra menggunakan 2 Line masing-masing membawa *bandwith* sebesar 2Mbps untuk operasional rumah sakit dan internet publik. Topologi yang akan di gunakan dapat di lihat pada gambar 4.2 berikut ini.



**Gambar 1. Topologi yang akan di gunakan**

4. *Implement*

Tahap *implement* adalah tahap-tahap penerapan terhadap rancangan jaringan dan rancangan yang telah di buat dan di setujui oleh Departemen Teknologi Informasi Rumah Sakit . Pada tahapan ini peneliti membagi tahap *peng-implimentasi*-nya menjadi beberapa bagian yaitu:

1. Instalasi Sistem Operasi Proxmox 3.4
2. Membuat *Virtual Machine* (VM) menggunakan KVM
3. Instalasi *Software-Software* yang akan di gunakan
4. Konfigurasi Mikrotik Routerboard

5. *Operate*

Merupakan fase di lakukannya uji coba dan pengoperasian sistem yang telah di bangun, adapun yang akan di uji adalah penggunaan *proxmox virtual Server* secara *Private* maupun secara *Public* serta pengujian-pengujian VM yang telah di bangun. Metode Pengujian yang digunakan adalah dengan metode *Blackbox testing* yang meliputi pengujian perangkat lunak yang berfokus pada fungsi *input* dan *output* aplikasi apakah sesuai dengan apa di harapkan atau tidak.

6. *Optimize*

Pada tahap terakhir dari metode PPDIOO adalah *Optimize*, yang meliputi perawatan maupun perbaikan terhadap sistem yang telah dibangun. Tahap pemeliharaan dan perbaikan ini terdapat beberapa proses yang harus di lakukan agar sistem dapat beradaptasi dengan perkembangan kebutuhan sistem jaringan,

**5. IMPLEMENTASI**

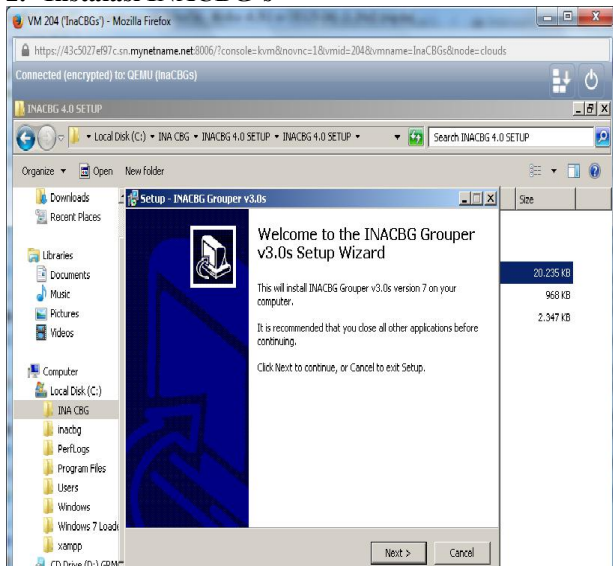
Pada tahapan implementasi peneliti membagi tahap *peng-implimentasi*-nya menjadi beberapa bagian yaitu:

## 1. Instalasi Sistem Operasi Proxmox VE 3.4



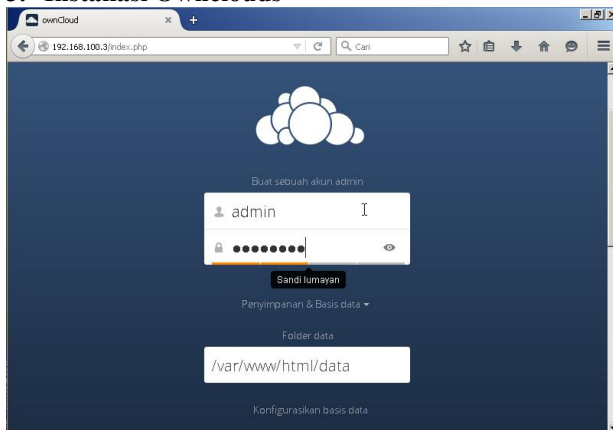
Gambar 2. Instalasi Proxmox

## 2. Instalasi INACBG's



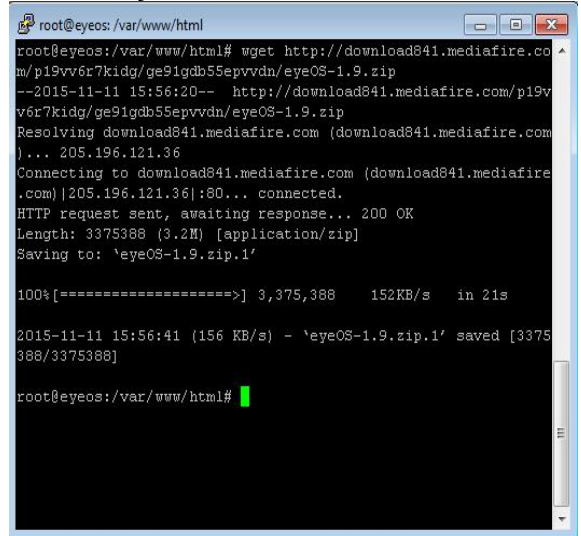
Gambar 3. Instalasi INACBG's

## 3. Instalasi Ownclouds



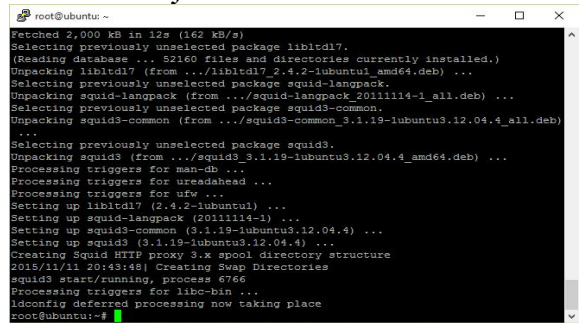
Gambar 4. Instalasi Owncloud

## 4. Instalasi EyeOS



Gambar 5. Instalasi EyeOS

## 5. Instalasi Proxy Server



Gambar 6. Instalasi Proxy Server

## 6. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka peneliti mengambil kesimpulan bahwa:

1. *Cloud Computing* dapat diterapkan di lingkungan kerja Rumah Sakit Samarinda Medika Citra.
2. *Cloud Computing* dapat dibangun menggunakan sistem operasi turunan Debian yaitu dengan menggunakan OS Proxmox 3.4
3. Jumlah kebutuhan *server* fisik di dalam lingkungan Rumah Sakit Samarinda Medika Citra dapat digantikan dengan menggunakan *Server* Virtualisasi yang telah dibangun.
4. Infrastruktur Teknologi Informasi dan Komunikasi yang dimiliki Rumah Sakit Samarinda Medika Citra saat ini telah mendukung implementasi tiga model layanan *Cloud Computing* yaitu *Private Cloud*, *Public Cloud* dan *Hybrid Cloud* untuk saat ini hanya menggunakan *Private Cloud* dikarenakan keterbatasan Bandwidth Internet.

5. Dengan adanya *Cloud Computing* arsitektur virtualisasi yang di hasilkan akan sangat bermanfaat dalam penghematan biaya perawatan, pengembangan serta implementasinya.
6. Perangkat lunak *open source* dapat di gunakan untuk memenuhi kebutuhan sistem informasi Rumah Sakit Samarinda Medika Citra

## 7. SARAN

Dalam penelitian peneliti menyadari masih banyak kekurangan, saran untuk Rumah Sakit Samarinda Medika Citra adalah:

1. Mengupgrade peralatan jaringan yang mendukung kecepatan sampai dengan 1 Gbps (*Gigabyte per second*).
2. Sewa *IP* Publik Statik, agar dalam perawatan dapat di lakukan di luar jaringan.
3. Mengupgrade *Bandwith* internet
4. Menambah kapasitas RAM (*Random Acces Memory*) pada *Server Cloud*
5. Menambah kapasitas *Hardisk* yang lebih besar pada *Server Cloud*

## 8. DAFTAR PUSTAKA

### Buku:

- Badiyanto, 2013, Cara Mudah Membangun Aplikasi Web PHP, Yogyakarta: Mediakom.
- Binanto Iwan, 2015, Sistem Operasi, Yogyakarta : Andi.
- Komputer, Wahana, 2008, Administrasi Jaringan Menggunakan Linux Ubuntu, Semarang: Andi
- Kurniawan, Wiharsono, 2007, Jaringan Komputer, Yogyakarta: Andi.
- Madcoms, 2009, Membongkar Misteri Internet Yogyakarta: Andi Publisher.

Pratama I Putu Agus Eka, 2014, *Smart City* Beserta *Cloud Computing* dan Teknologi-teknologi Pendukungnya, Bandung : Informatika

Pressman, Roger S. 2007. Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku I), Yogyakarta : Andi.

*Proxmox Virtual Envirotment*, <http://www.proxmox.com/proxmox-ve> di akses pada tanggal 30 Maret 2015

Purbo Onno W, 2008, Panduan Mudah Merakit dan Menginstal *Server Linux*, Yogyakarta: Andi.

Purbo Onno W, 2012, Membuat sendiri *Cloud Computing Server* Menggunakan *Open Source*, Yogyakarta: Andi Publisher

Sofana Iwan, 2012, Cisco CCNP dan Jaringan Komputer, Bandung: Informatika.

Sopandi, Dede, 2008, Instalasi dan Konfigurasi Jaringan Komputer, Bandung: Informatika.

Towidjojo Rendra, 2013, Mikrotik Kung Fu Kitab 1, Jasakom

### Artikel dari situs internet:

*EyeOS*, <http://www.eyeos.com> di akses pada tanggal 11 September 2015

INACBG'S, [http://www.buk.kemkes.go.id/sub\\_download.php?id=31](http://www.buk.kemkes.go.id/sub_download.php?id=31)

*Linux Journal*, <http://www.linuxjournal.com/content/eyeos-web-based-desktop-os> diakses pada 9 Nopember 2015

Owncloud, <https://www.owncloud.org/download>