

DESIGN DAN SIMULASI ARSITEKTUR SOFTWARE-DEFINED NETWORK/OPENFLOW UNTUK JARINGAN APLIKASI MANAJEMEN PENANGGULANGAN PASCA BENCANA

Rikie Kartadie¹⁾, Tommi Suryanto²⁾, Arif Syam³⁾

^{1), 2), 3)} Magister Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : rikie.kartadie@gmail.com¹⁾, tommi_suryanto2000@yahoo.com²⁾
syam_arif@yahoo.com³⁾.

Abstrak

Secara nasional upaya penanggulan bencana harus dilakukan secara komprehensif dan sistematis. Penyampaian informasi dari tiap shelter diharapkan dapat memberikan andil besar dalam proses penanggulangan pasca bencana. Software-Defined Network/Openflow disimulasikan dengan menguji tingkat efektifitas dengan kontrol langsung dari pusat. Tujuan diterapkannya SDN/Openflow pada jaringan penanganan pasca bencana adalah dapat meningkatkan nilai informasi yang diperoleh. Penerapan SDN/openflow pada jaringan diharapkan akan meningkatkan performa jaringan, mengurangi waktu dan biaya perawatan pada jaringan sehingga informasi dari shelter kepusat dapat berjalan dengan baik dan berkesinambungan.

Kata kunci: SDN/Openflow, penanganan pasca bencana, simulasi mininet.

1. Pendahuluan

Secara nasional upaya penanggulan bencana harus dilakukan secara komprehensif dan sistematis, namun dua hal ini masih terkendala dua masalah utama yaitu; belum memadainya kinerja aparat, kelembagaan penanggulangan bencana dan masih rendahnya kesadaran terhadap resiko bencana dan pemahaman terhadap kesiapsiagaan dalam menghadapi bencana.[1].

Memperhatikan situasi tersebut maka perlu adanya upaya-upaya meningkatkan pemahaman terhadap kesiapsiagaan dalam menghadapi bencana sehingga dapat meningkatkan pelayanan terhadap korban bencana, serta dapat meningkatkan penyampaian informasi yang diperlukan baik informasi yang diberikan dari shelter bencana kepusat dan sebaliknya dari pusat keshelter. Pelaksanaan program penanggulangan bencana ini, diperlukan data dan informasi aktual yang tersedia dengan cepat, tepat dan akurat. Untuk itu perlu dibangun pusat informasi penanggulangan bencana untuk memudahkan penyebaran informasi sehingga dapat digunakan dan dimanfaatkan oleh masyarakat baik secara

kelompok maupun perorangan guna mengetahui data dan informasi yang berkaitan dengan dampak-dampak bencana serta sekaligus dapat memberikan saran / pendapat ataupun bantuan nyata sebagai wujud partisipasi dan kepedulian dalam membantu sesama.[2] sehingga diperlukan suatu rancang arsitektur jaringan yang memiliki peformance yang tinggi dalam penyampaian data dan dapat dikontrol dengan baik secara terpisah, tersentral, mengurangi waktu dan biaya perawatan untuk meningkatkan penyampaian informasi.

Software-define network merupakan paradigma baru yang menjanjikan keuntungan diantaranya (1)Sentralisasi manajemen dan kontrol jaringan dari multi vendor, (2)Automatisasi pengembangan dan manajemen, (3)Inovasi terus menerus dalam kemampuan dan servis jaringan tanpa harus melakukan konfigurasi ulang pada setiap perangkat, (4)Dapat diprogram dengan mudah oleh admin, pengembang software, dan pengguna dengan bahasa pemrograman yang relatif mudah dipahami, (5)Meningkatkan realibility dan keamanan jaringan dan manajemen jaringan secara otomatis, (6)Penerapan kebijakan jaringan secara seragam, dan mengurangi kesalahan, (7)Kontrol jaringan secara granular(kecil) dengan kemampuan untuk diaplikasikan secara komprehensif dan kebijakan yang luas pada sesi, pengguna, perangkat, dan aplikasi.[3].

Bentuk *Software-Defined Networking* (SDN) memungkinkan para peneliti, administrator dan operator untuk mengontrol jaringan mereka dengan perangkat lunak khusus dan menyediakan Application Programming Interface (API) terhadap tabel forwarding dari switch dari vendor yang berbeda. [4]

Secara garis besar penerapan SDN/openflow dalam jaringan manajemen penanggulangan bencana dirancang untuk membantu penyampaian informasi penanganan pasca bencana alam. Seperti informasi korban bencana, informasi orang hilang/ditemukan dari shelter ke pusat, sehingga dapat segera diambil keputusan penanganan bencana yang diperlukan.

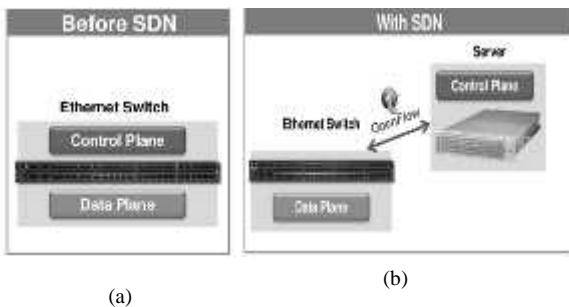
Tinjauan Pustaka

OpenFlow, adalah konsep yang sederhana, dimana

melakukan sentralisasi terhadap kerumitan dari jaringan kedalam sebuah software kontroler, sehingga seorang administrator dapat mengaturnya dengan mudah dengan hanya mengatur kontroler tersebut. McKeown, N., memperkenalkan konsep Open-Flow ini dengan ide awal adalah menjadikan sebuah network dapat di program/ dikontrol. [5]

Software defined networking (SDN) atau split arsitektur yang pertama kali diperkenalkan oleh Martin Casado di universitas stanford pada tahun 2007 adalah sebuah konsep yang memungkinkan/memperbolehkan operator jaringan untuk mengelola router dan switch secara fleksibel menggunakan software yang berjalan di server eksternal [6], yang dikutipnya dari IETF Working group, "Forwarding and control element separation (forces) framework,". Open Network Foundation, mendefinisikan SDN adalah sebuah Sebuah arsitektur network baru dimana kontrol jaringan dipisahkan dari forwarding dan diprogram secara langsung.[7]

Perbedaan antara perangkat tradisional dan perangkat yang menggunakan SDN/Openflow dapat dilihat pada gambar 1.

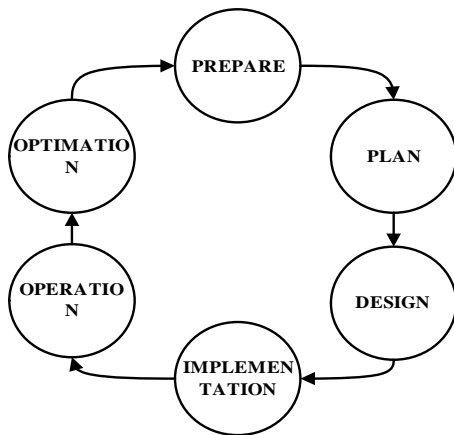


Gambar 1. (a) tanpa SDN, (b) dengan SDN

Topologi jaringan star memiliki Karakteristik ialah node berkomunikasi langsung dengan node lain melalui central node (hub/switch), traffic data yang mengalir dari node ke central node dan diteruskan ke node tujuan, jika salah [8]

Metodologi Penelitian

Metodologi penelitian yang digunakan sesuai dengan alur lifecycle seperti gambar 2.



Gambar 2. Design LifeCircle

Prepare (Persiapan);Tahap penetapan kebutuhan fungsional dan non fungsional pada jaringan manajemen pasca bencana, strategi pengembangan jaringan, usulan konsep arsitektur jaringan,dan identifikasi teknologi yang dapat memberikan support terbaik pada arsitektur.

Plan (Rencana); tahap perencanaan adalah tahapidentifikasi awal dalam kebutuhan dasar yang didasari pada tujuan akhir (hasil), fasilitas, dan kebutuhan pemakai(user).

Design (Perancangan);Pada tahap ini merupakan kelanjutan dari tahap Plan, yang dapat mengarahkan kita ke spesifikasi design yang komprehensif, design yang sesuai dengan kebutuhan teknis dan bisnis, dimana perancangan design disimulasikan dengan simulator.

Implement (implemetasi); Setelah design diterima, tahap implementasi (dan verifikasi) dapat dimulai. Jaringan atau komponen-komponen tambahan dibangun sejalan/sesuai.

Operate (operasi); Pengujian terakhir untuk kelayakan dari sebuah design.

Optimize (optimasi); Tahap optimasi adalah tahap yang melibatkan pihak mengidentifikasi manajemen dan yang menyelesaikan proaktif untuk isu-isu[8].

Kebutuhan non fungsional

Perangkat keras terdiri dari switch Openflow, Server Controller, dan sistem pengkabelan. Perangkat lunak terdiri dari, Operation Sistem GNU/Linux, Controller Open daylight, dan mininet simulator.

Kebutuhan fungsional

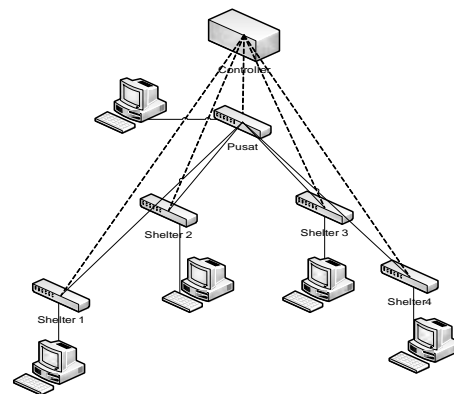
Terdiri dari tiga bagian utama, yaitu kehandalan, fleksibilitas, tingkat kesalahan yang rendah.

Kehandalan meliputi performance, bandwidth, manajemen security.

Arsitektur jaringan

Logical

Arsitektur logical pada perancangan SDN/Openflow pada jaringan penanggulangan pasca bencana dapat dilihat pada gambar 3.



Gambar 3. Arsitektur Logical

Layout Posisi Perangkat di Lapangan

Arsitekture layout posisi penempatan perangkat SDN/Openflow pada jaringan penanggulangan pasca bencana dilapangan dapat dilihat pada gambar 4.



Gambar 4. Layout Posisi Perangkat

2. Pembahasan

Instalasi Controller.

Dilakukan instalasi opendaylight v.0.1 dengan langkah seperti dibawah ini;

```
apt-get update
apt-get install maven git openjdk-7-jre
openjdk-7-jdk
git clone
http://git.opendaylight.org/gerrit/p/controller
r.git
cd
controller/opendaylight/distribution/opendaylight/
mvn clean install
cd target/distribution.opendaylight-0.1.0-SNAPSHOT-osgipackage/opendaylight
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-i386
```

Hasil dari intallasi controller dapat dilihat pada gambar 5 dibawah ini.



Gambar 5. Hasil intallasi controller

Design Topologi

Design topologi menggunakan mininet, coding menggunakan python, list coding seperti dibawah ini;

```
"""Custom topology example
SemnasTek Amikom Jogjakarta.
By Rikie, Tommi, Arif
Adding the 'topos' dict with a key/value pair
to generate our newly defined
topology enables one to pass in '--topo=mytopo'
from the command line.
"""
from mininet.topo import Topo
class MyTopo( Topo ): "Simple topology
example."
def __init__( self ):
"Create custom topo."
# Initialize topology
Topo.__init__( self )
# Add hosts and switches
topmidhost=self.addHost( 'h1' )
lefthost = self.addHost( 'h2' )
righthost = self.addHost( 'h3' )
bottomlefthost=self.addHost( 'h4' )
bottomrighthost=self.addHost( 'h5' )
leftswitch = self.addSwitch( 's2' )
midswitch = self.addSwitch( 's1' )
rightswitch = self.addSwitch( 's3' )
bottomleftswitch=self.addSwitch( 's4' )
bottomrightswitch=self.addSwitch( 's5' )
# Add links
self.addLink( lefthost, leftswitch )
self.addLink( leftswitch, midswitch )
self.addLink( midswitch, topmidhost )
self.addLink( midswitch, rightswitch )
self.addLink( rightswitch, righthost )
self.addLink( bottomrightswitch,
bottomrighthost )
self.addLink( bottomleftswitch, bottomlefthost )
self.addLink( bottomrightswitch, midswitch )
self.addLink( bottomleftswitch, midswitch )
topos = { 'mytopo': ( lambda: MyTopo() ) }
```

tampilan design topologi yang dirancang ditampilkan pada opendaylight



Gambar 6. Controller opendaylight

Design IP address pada topologi

Design IP yang digunakan pada perancangan jaringan SDN/Openflow seperti pada tabel 1

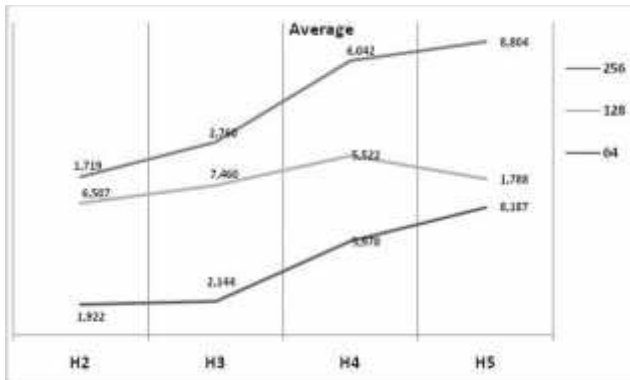
Tabel 1. Design IP Address

Nama	IP address	subnet
Controller	192.168.56.1:6633	24
Shelter Pusat	172.19.1.1	24
Shelter 1	172.19.2.1	24
Shelter 2	172.19.3.1	24
Shelter 3	172.19.4.1	24
Shelter 4	172.19.5.1	24

Simulasi Mininet

Pengujian dengan simulator mininet dengan melakukan pengujian koneksi antara shelter dan pusat serta melakukan pengujian penggunaan bandwidth pada masing-masing shelter, dari pengujian dihasilkan data-data sebagai berikut

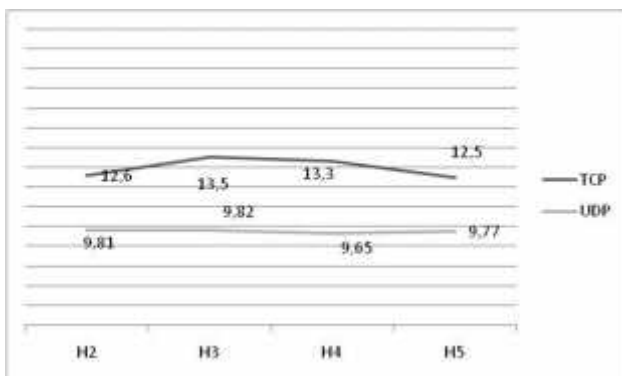
perbandingan pengujian koneksi dengan pengiriman paket data sebesar 64, 128 dan 256 byte pada masing-masing shelter dapat dilihat pada gambar 7.



Gambar 7. Rata-rata pengujian pengiriman paket data

Pengujian koneksi pada gambar 7 dapat dilihat bahwa transfer rate pada shelter 5 berbeda, hal ini dapat disebabkan TCAM tabel pada switch shelter 5 mengalami keterlambatan dalam loading time, namun tidak terjadi perbedaan yang signifikan yaitu 1,573 ms.

Hasil pengujian bandwidth pada masing-masing shelter dapat dilihat pada gambar 9.



Gambar 8. Pengujian Bandwidth

Pengujian diatas tampak bahwa besar bandwidth protokol TCP yang diuji memiliki bandwidth rata-rata sebesar 12,979 Mb/s dengan selisih bandwidth antar shelter sebesar 0,03 Mb/s..

3. Kesimpulan

Pengujian dengan melakukan pengujian koneksi antara shelter dan pusat serta melakukan pengujian penggunaan bandwidth pada masing-masing shelter dengan simulator mininet, dari pengujian koneksi dengan pengiriman paket data sebesar 64, 128 dan 256 byte memberikan response time rata-rata 4,903 ms dengan selisih tertinggi

pada shelter 5 yaitu 1.573 ms. Bandwidth rata-rata yang dapat diberikan untuk protokol TCP dan UDP masing-masing sebesar 12,975 Mb/s dengan selisih besar bandwidth tiap shelter rata-rata sebesar 0.03 Mb/s. Sehingga dapat dikatakan bahwa pemakaian arsitektur Software-Defined Network dapat memberikan performa yang lebih baik dan dapat melayani kebutuhan penyampaian data yang cepat pada jaringan Aplikasi Manajemen Penanggulangan Pasca Bencana.

Daftar Pustaka

- [1] BNPB, Rencana strategis badan nasional penanggulangan bencana 2010-2014, 2010
- [2] Kartadie, R., Asharudin, F., et al. APLIKASI SISTEM INFORMASI MANAJEMEN PENANGGULANGAN PASCA BENCANA, prosiding Seminar Nasional Teknologi Informasi dan Multimedia, ISSN 2302-3805, 2013
- [3] Anonim, Software-Defined Network: The New Norm for Network, ONF White Paper, 2013.
- [4] Applemen, M., De Boer, M., 30 Mei 2013, "Performance Analysis of OpenFlow Hardware", University Of Amsterdam, 2012
- [5] McKeown, N., et al., 1 Mei 2013, "OpenFlow: Enabling Innovation in Campus Networks", Stanford University, 2008
- [6] Shirazipour, M., John, W., Kempf, J., Green, H., Tatipamula, M., Realizing Packet-Optical Integration with SDN and OpenFlow 1.1 Extensions", Communications (ICC), IEEE International Conference, ISSN:1550-3607, 2012
- [7] Anonim, "Software-Defined Networking: The New Norm for Networks", Open Networking Foundation, 2012
- [8] Syahfrizal, Melwin, Pengantar Jaringan Komputer, Yogyakarta : ANDI, 2005
- [9] Tiso, J., Caswell, E., Designing Cisco Network Service Architecture (ARCH), 3rd ed., Cisco press, 2012.

Biodata Penulis

Rikie Kartadie, ST, Mendapatkan gelar Sarjana Teknik (ST) pada tahun 2001, sedang menempuh S2 Magister Teknik Informatika pada STMIK Amikom Yogyakarta konsentrasi Sistem Informasi semester 3, dalam tahap penyelesaian penulisan thesis, dan menjadi pengajar pada program studi S1 Teknik Informatika STMIK Amikom.

Tommi Suryanto, S.Kom, mendapatkan gelar Sarjana Komputer (Skom) pada tahun 2012, sedang menempuh S2 Magister Teknik Informatika pada STMIK Amikom Yogyakarta konsentrasi Teknologi Media Digital semester 3 dan dalam tahap penyelesaian penulisan thesis.

Arif Syam, S.Si., mendapatkan gelar Sarjana Sistem informasi (S.Si.) pada tahun 2010 di STMIK BINA BANGSA Kendari, sedang menempuh S2 Magister Teknik Informatika pada STMIK Amikom Yogyakarta konsentrasi Sistem Informasi semester 3 dan dalam tahap penyelesaian penulisan thesis