

Aplikasi Manajemen Jaringan Berbasis Software Defined Networking

Epaphras Ericson Thomas¹, Henry Palit², Agustinus Noertjahyana³
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp. (031)-2983455, Fax. (031)-8417658
E-mail: ericsn.th@gmail.com¹, hnpalit@petra.ac.id², agust@petra.ac.id³

ABSTRAK

Dalam sebuah korporasi besar yang memiliki banyak perangkat jaringan, seorang *network operator* seringkali harus melakukan konfigurasi masing-masing perangkat tersebut secara manual. Di sisi lain, adanya kebutuhan untuk mengorganisir ulang perangkat-perangkat tersebut menjadi lebih fleksibel sehingga dapat menjawab tantangan-tantangan seperti pembatasan akses *host* maupun *port* (*services*) tertentu.

SDN atau *Software Defined Network* diyakini sebagai salah satu solusi untuk menjawab tantangan diatas. Pada penelitian ini, sebagian masalah diatas akan dijawab menggunakan konsep dari SDN. Jika terdapat sebuah *point of management*, maka akan sangat dimungkinkan untuk dapat mengubah *network behaviour* (perilaku jaringan) dan atau konfigurasi lainnya, serta melakukan perubahan lain secara dinamis atau otomatis (diprogram), terutama dalam perubahan *policy*. Pada eksplorasi ini, OpenWRT ditanamkan pada perangkat jaringan sehingga dimungkinkan untuk dikontrol. Untuk memudahkan pemrograman jaringan, maka dibangun *web services* yang dilengkapi dengan otorisasi keamanannya.

Hasil dari eksplorasi ini menunjukkan bahwa konsep *Software Defined Networking* dapat menjawab masalah diatas, dengan biaya yang relatif murah.

Kata Kunci: OpenWRT, *software defined networking*, *web services*, konfigurasi jaringan, *network programming*.

ABSTRACT

In a large corporation with multiple network devices, a network operator often has to manually configure each device. However, there are some requirements to re-organize devices becomes more flexible in order to respond to challenges such as restricting host and port (services) access.

SDN or Software Defined Network is believed by some people to be one solution to answer the above challenges. In this study, some of the above problems will be answered using the concept of SDN. If there is a point of management, it will be possible to change network behavior and / or other configurations, and make other changes dynamically or automatically (programmed), especially in policy changes. In this exploration, OpenWRT is embedded in network devices so it is possible to be controlled. To facilitate the programming of the network, then built web services that are equipped with security authorization.

The results of this exploration show that the concept of Software Defined Networking can answer the above problems, with relatively low cost.

Keywords: *OpenWRT, software defined networking, web services, network configuration, network programming.*

1. PENDAHULUAN

Berada dalam suatu zaman yang sungguh mengedepankan kemajuan teknologi informasi, kebutuhan akan media internet sudah cukup tinggi, yang didukung juga dari pesatnya perkembangan teknologi infrastruktur jaringan oleh beberapa *vendor* atau merk. Banyaknya ragam perangkat keras tersebut akhirnya membuat pengelola jaringan (*network administrator*) harus melakukan pengaturan atau konfigurasi perangkat maupun logika jaringan (*control*) secara manual; walaupun biasanya setiap *vendor* memiliki ciri khas nya sendiri, bahkan tata cara konfigurasi yang relatif berbeda.

Pengaturan atau konfigurasi yang dilakukan oleh *network administrator* saat ini masih cenderung menerapkan sistem *low level configuration* atau konfigurasi secara manual pada setiap perangkat jaringan yang digunakan. Apabila pengelola jaringan memiliki sepuluh perangkat berupa *router*, maka mereka juga harus melakukan pengaturan yang cenderung sama secara berurutan dan bergantian dari satu perangkat ke perangkat lain. Hal ini sungguh menjadi masalah ketika perangkat yang digunakan cukup banyak dan terkadang bagi peneliti, hal tersebut terasa sangat membosankan dan kurang efektif walaupun itu merupakan tugas dan kewajiban pengelola jaringan.

Banyak waktu dan tenaga akhirnya terbuang begitu saja, bahkan pengaturan yang dilakukan manual secara bergantian memiliki potensi untuk terjadi kesalahan (*fault*) yang dapat membuat perangkat jaringan akhirnya menjadi tidak berfungsi secara maksimal; padahal akan sangat efisien dan memudahkan *network administrator* jika memiliki sebuah *point of management* untuk dapat mengatur dan mengelola ribuan perangkat jaringan sekaligus dalam waktu yang cukup singkat, kebutuhan mengenai hal ini berkaitan dengan *orchestration* dan *scalability*. Setiap perangkat yang ditambahkan dalam sistem akan memiliki *Application Programming Interface* dengan keluaran data berupa json, yang dapat diperoleh melalui *REST based web services* dan dapat diolah atau dianalisa melalui lintas bahasa pemrograman (*cross platform*).

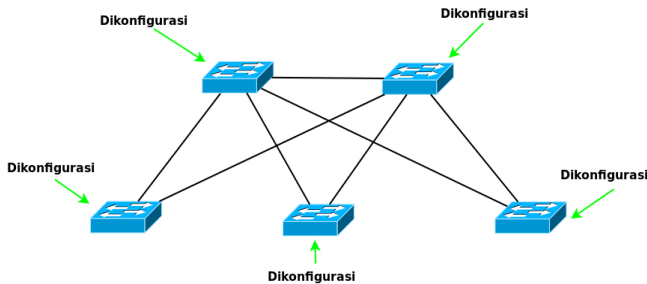
Seiring dengan berkembangnya teknologi perangkat jaringan, dunia jaringan internet sendiri akan terus mengalami perubahan dan perkembangan, ketika saat ini konsep internet masih harus dikonfigurasi secara konvensional, kelak di masa yang akan datang, sangat dimungkinkan bahwa konsep jaringan dapat dikombinasikan dengan konsep pemrograman menjadi hal baru berupa konsep *network programming*, bahkan mendukung aspek modularitas maupun *extensible programming*.

Melihat dari sisi opportunity perkembangan teknologi tersebut, tidaklah berlebihan jika ada sebuah aplikasi atau piranti lunak untuk dapat mengelola semua hal di atas, sehingga dari sisi infrastruktur jaringan serta sisi *control*-nya dalam perangkat dapat dimanfaatkan semaksimal mungkin; dan berangkat dari masalah ini aplikasi manajemen jaringan ini dirancang.

2. TINJAUAN PUSTAKA

2.1 Software Defined Networking

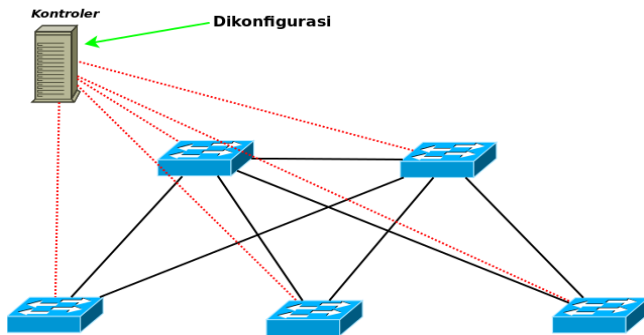
Konsep dalam *network layer* merupakan contoh dari abstraksi jaringan sebelum adanya konsep *Software Defined Networking*. Konsep *layer* ini juga hanya terbatas untuk meng-abstraksi-kan *data plane*, tidak ada konsep yang mewakili *control plane*. Sebenarnya tidak ada yang salah dengan ini, namun akan menjadi tugas tambahan apabila nanti jaringan sudah semakin kompleks seperti pada Gambar 1.



Gambar 1. Konsep jaringan tradisional

Dengan menggunakan metode ini, seiring dengan berjalannya waktu, nantinya akan semakin banyak mekanisme (protokol) yang menyebabkan jaringan berkembang semakin kompleks dan semakin sulit untuk dikonfigurasi atau dikelola, bahkan akan menghalangi dan memperlambat inovasi teknologi karena begitu kompleksnya mekanisme seperti Gambar 2.

Software Defined Networking bekerja dalam sebuah *controller* (*point of management*) yang sangat memungkinkan sebuah entitas aplikasi dapat mengendalikan seluruh proses komunikasi dalam beberapa resources jaringan, mengendalikan *traffic* yang melewati perangkat jaringan, serta melakukan suatu inspeksi khusus atau modifikasi terhadap jaringan yang melewatinya.



Gambar 2. Konsep jaringan modern (SDN)

Konsep *Software-Defined Network*, pertama kali diperkenalkan oleh Martin Casado di Universitas Stanford pada tahun 2007 dengan tulisan pada jurnalnya berjudul "*Ethane: Taking Control of the Enterprise*". Pada jurnal tersebut dijelaskan bahwa *ethane* merupakan suatu arsitektur baru untuk suatu perusahaan yang mengizinkan manajer mendefinisikan luasan dari sebuah jaringan, kebijakan didalamnya, serta menjalankan hal-hal tersebut secara

langsung. *Ethane* adalah sebuah konsep penyederhanaan yang cukup ekstrim dari *ethernet switch* dengan sebuah *controller* terpusat yang dapat mengatur hak masuknya data dan aliran *routing*.

Software-Defined Networking (SDN) adalah sebuah konsep baru yang memungkinkan pengelola jaringan untuk mengelola *router* dan *switch* secara fleksibel menggunakan software yang berjalan pada *server* eksternal. *Open Network Foundation*, mendefinisikan SDN adalah sebuah arsitektur jaringan baru di mana kontrol jaringan dipisahkan dari *forwarding plane* dan dapat diprogram secara langsung. *Software Defined Network* merupakan jaringan pintar yang memiliki sentralisasi dalam hal *logical software* (*software-base*), selain itu ONF menyatakan bahwa dengan SDN tidak lagi membutuhkan protokol standar, tetapi cukup hanya menerima instruksi dari sebuah *SDN controller*. [2].

2.2 OpenWRT

OpenWRT merupakan satu dari sekian banyak *distro linux* yang didesain secara khusus untuk tipe perangkat *embedded* [4]. Sistem operasi ini menyediakan *file system* dengan manajemen paket berupa *firmware*. OpenWRT merupakan sebuah kerangka atau pondasi dasar untuk pengembangan aplikasi, dimana kerangka ini memungkinkan pengguna untuk dapat menambahkan aplikasi tertentu sebagai penunjang kerja perangkat yang digunakan.

Ada banyak versi dari OpenWRT, setiap versi diberi nama minuman beralkohol, biasanya akan tampak pada tampilan setelah login terdapat resep dari minuman alkohol tersebut [5]. Untuk melakukan konfigurasi OpenWRT, dapat dilakukan melalui tampilan atau antarmuka website menggunakan web browser atau dapat juga diakses melalui mode teks (CLI) dengan *remote SSH*.

2.3 Python

Python adalah salah satu jenis bahasa pemrograman dengan orientasi objek dinamis untuk pengembangan berbagai macam perangkat lunak. Bahasa pemrograman ini memberikan dukungan yang cukup untuk dapat terintegrasi dengan bahasa pemrograman lain maupun pustaka pendukung standar lainnya (modul atau *library*) yang dapat dipelajari dengan mudah dan cepat.

Biasanya, para pengembang menggunakan bahasa pemrograman *Python* karena mereka mendapatkan nilai produktivitas yang lebih tinggi daripada bahasa pemrograman lain, setiap kode sumber yang mereka ciptakan dapat terus dipelihara dalam jangka waktu relatif panjang (*update support*). Selain itu, bahasa pemrograman *Python* dapat berjalan dalam sistem operasi umumnya seperti *windows*, *linux/unix*, *mac os x*, *amiga*, hingga *palm handhelds* perangkat ponsel milik nokia.

Perkembangan *Python* saat ini sudah semakin maju, *Python* telah diporting dalam mesin *virtual* ternama seperti *Java* maupun *.NET* dan saat ini telah didistribusikan secara resmi dibawah lisensi *opensource* yang disetujui oleh lembaga OSI (*Opensources Initiatives*), sehingga membuat *Python* bebas digunakan dan gratis walaupun untuk kepentingan komersil [1].

2.4 Bash Shell

Merupakan singkatan dari *Bourne Again Shell*, *Bash Shell* merupakan turunan dari *Bourne Shell* (*sh*), dan telah menjadi standar *shell* di sistem operasi *Linux*. *Bash* memiliki beberapa keunggulan seperti pengeditan baris perintah secara terstruktur, *command completion* dan pemanggilan ulang perintah yang pernah diketik dalam *prompt*. Selain itu, *Bash* juga merupakan

bahasa pemrograman yang terstruktur, dimana struktur pembuatan programnya jelas berdasarkan langkah yang diperlukan [3].

2.5 Laravel

Merupakan salah satu *framework* yang menggunakan dasar bahasa pemrograman PHP dengan orientasi pada nilai-nilai kesederhanaan dan fleksibilitas pada sisi desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di *repository Github*. Sama seperti *framework* PHP lainnya seperti *CodeIgniter* atau *Yii*, Laravel dibangun dengan basis MVC (*Model-View-Controller*). *Framework* laravel dilengkapi dengan sebuah *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* maupun instalasi *bundle*.

Framework ini dibuat oleh Taylor Otwell, dan pembuatannya dimulai pada April tahun 2011; proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang *up-to-date* dengan versi PHP. Mengembangkan *framework* yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya didalamnya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri *framework* dengan nama Laravel dengan syarat minimum penggunaan PHP versi 5.3 keatas.

Laravel merupakan *framework* yang bagus untuk aplikasi berbasis *website* yang mendukung aspek modularitas dan fleksibilitas, dan mendukung nilai keamanan aplikasi yang cukup tinggi serta dukungan fitur otentikasi, *routing*, *session*, dan *caching*. *Framework* ini berusaha untuk menggabungkan yang terbaik dari apa yang ada pada *framework* web lain, termasuk *framework* yang menggunakan bahasa pemrograman lain, seperti *Ruby on Rails*, *ASP.NET MVC*, dan *Sinatra*.

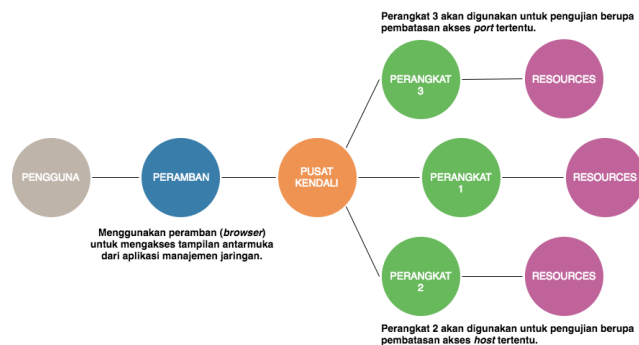
Laravel juga mudah diakses, *powerful* dan menyediakan *tools* yang diperlukan untuk *level* aplikasi cukup besar dan dapat terintegrasi dengan beberapa aplikasi lainnya.

3. ANALISIS DAN DESAIN

3.1 Desain Konseptual

Aplikasi manajemen ini dirancang secara khusus dengan tampilan antarmuka berbasis *website* menggunakan *framework* Laravel 5.1 dengan bahasa pemrograman *PHP*; *MySQL* sebagai aplikasi penyimpanan data *database* untuk penyimpanan beberapa *data* maupun untuk keperluan *export* dan *import* data perangkat maupun konfigurasi di dalamnya. Dalam aplikasi ini,terdapat tiga modul utama yaitu menu *device*, *scheduler* dan *config*.

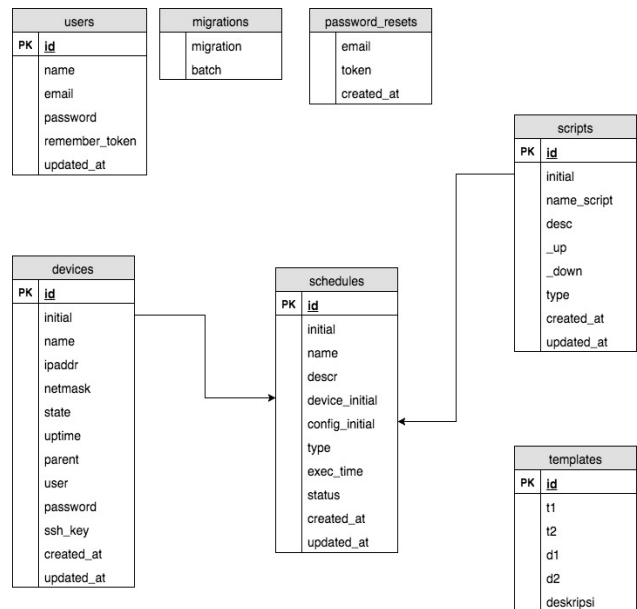
Gambar 3 dibawah ini merupakan sebuah diagram konseptual yang menggambarkan konsep dan keterhubungan antar perangkat, baik perangkat jaringan maupun perangkat kendali (*controller*).



Gambar 3. Desain konseptual

3.2 Desain Database

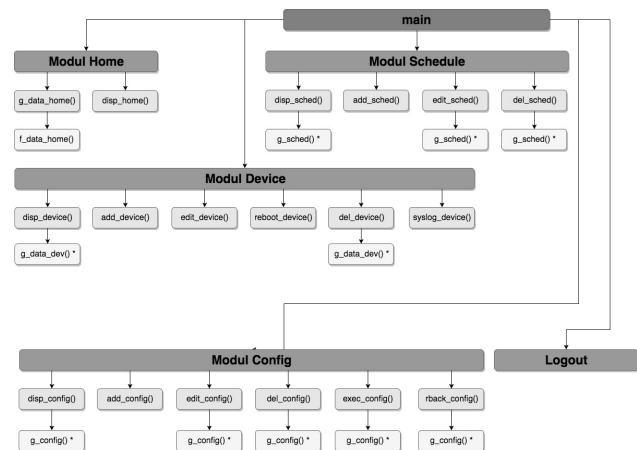
Pada Gambar 4, merupakan desain tabel yang digunakan untuk membangun aplikasi manajemen jaringan menggunakan konsep *Software Defined Network*



Gambar 4. Desain tabel database

3.3 Diagram Alir Menu

Berikut ini merupakan sebuah diagram untuk menjelaskan menu yang dimiliki oleh aplikasi manajemen jaringan ini. Dalam setiap modul terdapat fungsi yang beragam seperti *edit*, *delete* dan lain sejenisnya, yang dapat dilihat pada Gambar 5.



Gambar 5. Diagram alir menu aplikasi

3.4 Inovasi dan Pemanfaatan SDN

Berkembangnya teknologi perangkat keras (*hardware*) jaringan yang begitu pesat, memberikan dampak dan tantangan tersendiri untuk mengatur dan mengelola perangkat tersebut lebih kompleks. Adanya kebutuhan inovasi dalam bidang jaringan yang semakin kompleks didasari oleh beberapa masalah, beberapa masalah yang dimaksud adalah sebagai berikut:

- Semakin banyaknya infrastruktur atau perangkat jaringan yang digunakan, membuat *network administrator* harus

melakukan konfigurasi masing-masing perangkat tersebut secara manual; padahal akan sangat efisien dan memudahkan jika memiliki sebuah *point of management* untuk dapat mengatur dan mengelola bahkan ribuan perangkat sekaligus dalam waktu yang cukup singkat, kebutuhan ini berkaitan dengan *orchestration* dan *scalability*.

- Tidak dapat dipungkiri, sebagian besar pengaturan yang dilakukan oleh *network administrator* cenderung bersifat manual, dalam kutip bahwa *troubleshooting*, perubahan *policy* dan sejenisnya dilakukan secara manual satu demi satu. Sangat memudahkan jika dalam sebuah *point of management* memiliki sebuah kemampuan untuk mengubah *network behaviour* (perilaku jaringan) dan juga melakukan perubahan-perubahan tersebut secara otomatis, terutama dalam perubahan *policy*.
- Kebutuhan untuk dapat *memonitor* jaringan baik dari sisi sumber daya, konektivitas dan lainnya dalam sebuah *point of management*.

Software Defined Network lebih mengarah pada sisi fleksibilitas dan modularitas dengan membawa konsep *modular programming* menuju dunia *networking*, sehingga dapat membuat fungsionalitas perangkat menjadi lebih kompleks, mudah, cepat dan handal tanpa harus mengoperasikan protokol baru. Ada beberapa aspek penting dari SDN, yaitu sebagai berikut:

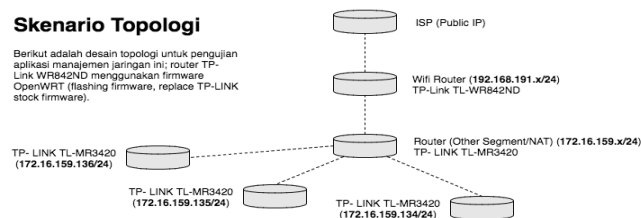
- Adanya pemisahan secara fisik atau eksplisit antara *forwarding data-plane* dan *control plane*,
- Antarmuka *standard (vendor agnostic)* untuk memprogram perangkat jaringan,
- *Control-plane* yang terpusat (secara logika) atau adanya sistem operasi jaringan yang mampu membentuk peta logika (*logical map*) dari seluruh jaringan dan kemudian memrepresentasikannya melalui (sejenis) API (*Application Programming Interface*),
- Virtualisasi di mana beberapa sistem operasi jaringan dapat mengontrol bagian-bagian (*slices* atau *substrates*) dari perangkat yang sama.

3.5 Metode Komunikasi

Untuk dapat berkomunikasi dengan perangkat, aplikasi akan menggunakan protokol *SSH* yang sudah ada dalam sistem operasi OpenWRT, tanpa harus melakukan instalasi lain sejenisnya. Pertama kali, sistem akan *login* menggunakan *username (root)* dan *password* yang telah diberikan oleh pengguna, sistem akan menciptakan sebuah *public key* melalui *engine* menuju perangkat akan menggunakan *ssh* secara *remote*. Perintah dalam melakukan konfigurasi akan mengikuti standart baku milik OpenWRT, yaitu *UCI (Unified Configuration Interface)*.

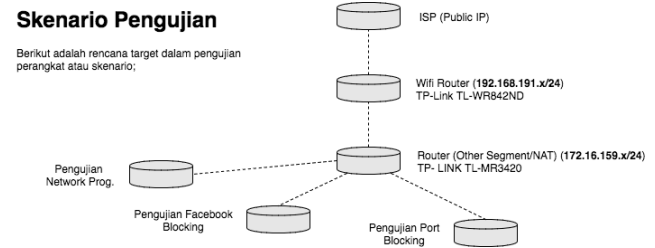
3.6 Skenario Topologi dan Pengujian

Skenario topologi yang digunakan dalam eksplorasi ini dapat dilihat pada Gambar 6.



Gambar 6. Skenario topologi

Sedangkan untuk skenario pengujian yang digunakan dalam eksplorasi ini dapat dilihat pada Gambar 7.



Gambar 7. Skenario pengujian

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Skenario

Bagaimana cara untuk mengotomasikan pengaturan jaringan dalam banyak perangkat jaringan (*router*) secara massal dengan satu kali pengaturan atau konfigurasi ?

- Otomasi pengaturan jaringan menuju banyak perangkat dapat dilakukan dengan memilih *checkbox* perangkat pada bagian *scheduler*; satu konfigurasi dapat digunakan untuk banyak perangkat jaringan dalam waktu yang relatif singkat secara bersamaan. Konfigurasi yang diikat dengan waktu eksekusi milik *scheduler* akan dikirimkan melalui protokol *ssh* menuju perangkat tujuan sesuai data *scheduler*.

Bagaimana cara atau implementasi teknis untuk menggabungkan konsep *networking* dan *programming* menjadi *modular network programming* ?

- Melakukan kombinasi antara *networking* dan *programming* sudah dapat dilakukan karena hadirnya dukungan *Application Programming Interface* atau API pada setiap perangkat; jadi sisi kontrol atau manajemen dapat dilakukan oleh perangkat jaringan lintas *vendor* dan versi, disini nilai *orchestration* juga bermain. *Network* yang dapat diprogram akan sangat memudahkan pengelola jaringan, misalnya melakukan pengaturan otomatis pada saat tertentu, melakukan *restart* perangkat pada jam tertentu, bahkan untuk melakukan *blocking host* seperti *facebook* pada jam kerja dan lain sebagainya.

Konsep Software Defined Networking dan Orchestration.

- OpenWRT merupakan salah satu distribusi atau distro linux untuk digunakan pada perangkat embedded. Karena ukuran filesystem yang terbatas, membuat beberapa perintah atau command yang dimiliki juga terbatas. Ukuran rom atau memory yang digunakan oleh OpenWRT sangat menyesuaikan dengan perangkat jaringan, dengan kata lain untuk setiap seri atau keluaran perangkat baru oleh vendor yang sama sekalipun, akan terdapat perbedaan jika setiap seri dan teknologi perangkat jaringan saling dibandingkan. *Sinatra*.
- OpenWRT sendiri menggunakan shell khusus yang dikemas dalam paket khusus bernama BusyBox, dengan shell script (*ash shell*). Berbeda dengan distro linux lainnya seperti ubuntu, debian dan lain sejenisnya, distro OpenWRT tidak dapat menangani pemrosesan atau mengerjakan kebanyakan perintah (*command*), seperti perintah *resolveip*, atau *host* (untuk melakukan resolve domain atau hostname ke ip address). Maka dari itu, perlu ada sebuah tools atau modul tambahan untuk dapat mengatasi kekurangan tersebut.

- Software Defined Networking, secara gamblang dapat dijelaskan sebagai sebuah piranti lunak yang dapat mendefinisikan segala fungsi kontrol jaringan.

```
if commander[0] == "#!/custom":
    if commander[1] == "blocker.sh": #Blocker Script
        try:
            s = requests.get(
                'http://mrc.eric.dev/api/devices/'+
                initial['device_initial'] + '/get?token=' +
                TOKEN
            )
            info = s.json()
```

Contoh eksekusi :

```
./blocker.sh <hostname> <iface> <port>
```

- Bagian ini, dapat digunakan sebagai kamus perintah atau command, satu perintah, dapat di definisikan untuk berbagai macam jenis perangkat jaringan, kedepannya.

5. KESIMPULAN

Dari hasil pengujian dan penelitian terkait dengan dua rumusan masalah diatas, maka aplikasi ini sudah cukup bermanfaat dan membantu para pengelola jaringan dalam memantau, memelihara perangkat jaringan yang dimiliki, serta memungkinkan untuk terjadinya orkestrasi perangkat jaringan; dimana pengaturan dalam sisi manajemen perangkat dapat dilakukan dengan satu bahasa, walaupun dengan menggunakan perangkat jaringan yang berbeda sekalipun.

Pemrograman jaringan juga sudah sangat dimungkinkan, dengan menggunakan konsep pertukaran data standar, maka pengelola

jaringan akan lebih mudah dalam menggunakan data tersebut. Data yang dihasilkan dapat dianalisis ataupun sebagai sumber data untuk aplikasi pendukungnya. Dengan kata lain, aplikasi ini tidak menutup kemungkinan untuk diperbesar ataupun ditambahkan modul-modul khusus untuk memaksimalkan kinerja perangkat maupun *controller* ini.

6. DAFTAR REFERENSI

- [1] Harismawan, A. F. dan Kharisma, A. P. 2018. *Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Vol. 2.
- [2] Kartadie, R. dan Satya, B. 2015. *Uji Performa Implementasi Softwar-Based Openflow Switch Berbasis OpenWRT Pada Infrastruktur Software-Defined Network*. Jurnal Ilmiah Universitas Amikom, Vol 16, No 3.
- [3] Milzam. 2014. *Pengembangan Sistem Operasi Linux Untuk Keamanan Jaringan*. Jurnal STIMIK U'BUDIYAH Indonesia.
- [4] Prasetyo, A. B. 2013. *Implementasi Portabilitas VOIP menggunakan Asterisk Pada OpenWRT*. Jurnal Universitas Dian Nuswantoro, 3.
- [5] Rikhmawan, A. 2017. *Memasang OpenWRT di Huawei HG 553 dan Modem Prolink PHS101*. URI: <https://agengr.akjepara.ac.id/memasang-openwrt-di-huawei-hg-553-dan-modem-prolink-phis101/>.